Mário Rui Monteiro Marques

Master of Science in Electrical Engineering

Reference Model for Interoperability of Autonomous Systems

Dissertação para obtenção do Grau de Doutor em
Engenharia Eletrotécnica e de Computadores

Orientador: Fernando Coito,
Prof. Associado,
Universidade Nova de Lisboa

Co-orientador: Victor Lobo,
Prof. Catedrático,
Escola Naval

Júri:

Presidente: Prof. Doutor Jorge Teixeira, FCT-UNL

Arguentes: Prof. Doutor José Victor, IST
Prof. Doutor António Serralheiro, AM

Vogais: Prof. Doutor Jorge Lobo, UC
Prof. Doutor Aníbal Matos, FEUP
Prof. Doutor José Oliveira, FCT-UNL
Prof. Doutor Fernando Coito, FCT-UNL

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Dezembro, 2018

Reference Model for Interoperability of Autonomous Systems

To Ana, Martim e Mariana for their love and full support

Acknowledgements

Firstly, I would like to thank my supervisor, Professor Fernando Coito, and co-supervisor, Professor Victor Lobo for their guidance, patience and contribution to the successful completion of this thesis work.

My acknowledgments to the Portuguese Navy for giving me the opportunity to be a PhD student and to FCT for hosting this research.

My appreciation to the Professors and staff of the Naval Academy for their support.

I would like to thank the constant support of my loving wife Ana and my son Martim and daughter Mariana, for their love, motivation, patience and understanding that allowed me to finish this challenge.

Resumo

Esta tese propõe um modelo de referência para descrever os componentes de um Sistema Não-Tripulado Aéreo, de Superfície ou Subaquático (UxS) e o uso de um *Sistema para Alcançar Interoperabilidade* (IBB) para comandar, controlar e obter feedback de tais veículos. A importância e as vantagens de tal modelo de referência, com uma nomenclatura padrão e taxonomia, são mostradas. Analisamos os conceitos de interoperabilidade e alguns esforços para alcançar modelos de referência comuns em outras áreas. Em seguida, apresentamos uma visão geral dos sistemas não-tripulados existentes, descrevendo a sua história, características, classificação e missões. O conceito de *Interoperability Building Blocks* (IBB) é apresentado para descrever padrões, protocolos, modelos de dados e *frameworks*, e um grande conjunto deles é analisado. Um novo e poderoso modelo de referência para o UxS, denominado RAMP, é proposto, que descreve os vários componentes que um UxS pode ter. É um modelo hierárquico com quatro níveis, que descreve as componentes do veículo, a ligação de dados e o segmento terrestre. O modelo de referência é validado mostrando como ele pode ser aplicado e pode estruturar a descrição e o modo como os UxS são usados em vários projetos em que o autor trabalhou. Um exemplo é dado sobre como um único padrão foi capaz de controlar um conjunto heterogéneo de UAVs, USVs e UGVs.

Keywords: Veículos Autónomos,Modelo de Referência,Interoperabilidade.

Abstract

This thesis proposes a reference model to describe the components of an Unmanned Air, Ground, Surface, or Underwater System (UxS), and the use of a single Interoperability Building Block to command, control, and get feedback from such vehicles. The importance and advantages of such a reference model, with a standard nomenclature and taxonomy, is shown. We overview the concepts of interoperability and some efforts to achieve common reference models in other areas. We then present an overview of existing unmanned systems, their history, characteristics, classification, and missions. The concept of Interoperability Building Blocks (IBB) is introduced to describe standards, protocols, data models, and frameworks, and a large set of these are analyzed. A new and powerful reference model for UxS, named RAMP, is proposed, that describes the various components that a UxS may have. It is a hierarchical model with four levels, that describes the vehicle components, the datalink, and the ground segment. The reference model is validated by showing how it can be applied in various projects the author worked on. An example is given on how a single standard was capable of controlling a set of heterogeneous UAVs, USVs, and UGVs.

Keywords: Unmanned Systems, Reference Model, Interoperability.

Contents

List of Figures

List of Tables

Acronyms

A

ABE - Autonomous Benthic Explorer

ACM - Association for Computing Machinery

ADCP - Acoustic Doppler Current Profiler

ADS – B - Automatic Dependent Surveillance–Broadcast

AHRS - Attitude and Heading Reference System

AIS - Automatic Identification System

ALV - Autonomous Land Vehicle

ASC - Application Software Component

ASCAMM - Associació Catalana d'Empreses constructores de Motlles i Matrius

ASI - Air Speed Indicator

ASISTS - Aircraft Ship Integrated Secure and Traverse System

ASW - Anti-Submarine Warfare

ATEX – Atmosphere Explosive

AT/FP - Anti-Terrorism / Force Protection

AV – Air Vehicle

AVCL - Autonomous Vehicle Command Language

AUSS - Advanced Unmanned Search System

AUV – Autonomous Underwater Vehicle

B

BEAR - Battlefield Extraction Assist Robot

BLOS - Beyond Line Of Sight

BML - Battle Management Language

BSD - Berkeley Software Distribution

C

C2 - Command and Control

C2IS - Command and Control Information Systems

C4I - Command, Control, Communication, Computer and Intelligence

CAE - Computer Aided Engineering

CBRN - Chemical, Biological, Radiological and Nuclear

CCI - Command and Control Interface

CCITT – Comité Consultatif International Téléphonique et Télégraphique

CCISM - Command and Control Interface Specific Module

CCU - Command and Control Unit

CDT - Control Data Terminal

CINAMIL - Portuguese Army Research Center

CINAV - Portuguese Navy Research Center

CLARAty - Coupled Layered Architecture for Robotic Autonomy

CMRE - Centre for Maritime Research and Experimentation

CN3 - Communication/Navigation Network Codes

CNAD - Council of NATO National Armaments Directors

CommonCL – Common Compact Language

CompactCL – Compact Control Language

CONOPS – Concept of Operations

CRC - Cyclic Redundancy Check

CS - Control Station

CSD - Coalition Shared Database

CSO - Collaboration Support Office

CSW - CRITICAL Software

CUCS – Core UAV Control System

D

DARPA - Defense Advanced Research Projects Agency

DCCL - Dynamic Compact Control Language

DDS - Data Distribution Service

DIN - Deutsches Institut für Normung

DLI – Data Link Interface

DM – Domain Model

DoD – Department of Defense

DVL - Doppler Velocity Log

E

ECOA - European Component Oriented Architecture

EO – Electro Optics

EOD - Explosive Ordnance Disposal

EPnP - Efficient Perspective n Point

ESM – Electronic Warfare Support Measures

ET - Exploratory Teams

ETC2 – Estação Terrestre Comando e Controlo

ETP – Estação Terrestre Payload

ETHZ - Eidgenoessische Technische Hochschüle Zürich

EW - Electronic Warfare

F

FEUP - Faculdade de Engenharia da Universidade do Porto

G

GCS - Ground Control Station

GIS - Geographic Information System

GMDSS - Global Maritime Distress and Safety System

GMTI - Ground Moving Target Indicator

GPL - General Public License

GPS – Global Positioning System

GSM - Global System for Mobile Communications

GUI - Graphical User Interface

H

HCI - Human Computer Interface

HMI - Human Machine Interface

HLD - Homeland Defense

I

IA - Influence Activities

IBB - Interoperability Building Blocks

IBM - International Business Machines

ICARUS - Integrated Components for Assisted Rescue and Unmanned Search Operations

ID - Inspection/Identification

IEEE – Institute of Electrical and Electronics Engineers

IMC - Inter-Module Communication

IMU - Inertial Measurement Unit

INESC - TEC - Institute for Systems and Computer Engineering, Technology and Science

INS - Inertial Navigation System

IoC – Inversion of Control

IR- Infra Red

ISO – International Organization for Standardization

ISR – Intelligence, Surveillance and Reconnaissance

ITIL – Information Technology Infrastructure Library

ITSM - IT Service Management

ITU - International Telecommunication Union

J

JAUGS - Joint Architecture for Unmanned Ground Systems

JAUS - Joint Architecture for Unmanned Ground Systems

JC3IEDM - Joint Consultation, Command and Control Information Exchange Data Model

JCIDS - Joint Capabilities Integration Development System

JSD - JAUS Service Definition

JSIDL - JAUS Service Interface Definition Language

L

LAN - Local Area Network

LARS - Launch and Recovery System

LAUV - Light Autonomous Underwater Vehicle

LBL - Long Base Line

LCGLE - Land Capability Group for Land Engagement

LCM - Lightweight Communications and Marshaling

LDM - Logical Data Model

LGPL - Lesser General Public License

LHLRS - Light Harpoon Landing Restraint System

LIDAR - Light Imaging Detection and Ranging

LOI - Levels of Interoperability

LORAN - Long Range Navigation

LOS - Line Of Sight

LS - Lecture Series

LUGV - Large UGV

LVDT - Linear and Rotary Variable Differential Transformer

M

MAC - Medium Access Control

MAJIIC - Multi-sensor Aerospace-ground Joint Intelligence surveillance and reconnaissance Interoperability Coalition

MAV - Micro Air Vehicle

MAVlink - Micro Air Vehicle communication protocol

MB - Main Blocks

MCI - Multilateral Interoperability Programme Common Interface

MCM - Mine Countermeasures

MDARS - Mobile Detection Assessment and Response System

MDCS - Multi-Domain Control Station

MDPU - Medium access control Protocol Data Units

MFW - Multi-Function Workstation

MIP - Multilateral Interoperability Programme

MIT – Massachusetts Institute of Technology

MIO - Maritime Interdiction Operations

MS - Main Systems

MSG - Message

MOOS - Mission Oriented Operating Suit

MOOSApp - MOOS Applications

MOOSDB – MOOS Database

MOU - Memorandum of Understanding

MPCS - Mission Planning and Control System

N

NAAG - NATO Army Armaments Group

NASA – National Aeronautics and Space Administration

NATO - North Atlantic Treaty Organization

NCOIC - Network Centric Operations Industry Consortium

NIAG - NATO Industrial Advisory Group

NNAG - NATO Naval Armaments Group

NPS - Naval Postgraduate School

O

OGC - Open Geospatial Consortium

OMG - Object Management Group

OODA - Observe, Orient, Decide and Act

OSI - Open System Interconnection

OSRF - Open Source Robotics Foundation

OSOCC - On-Site Operations and Coordination Center

OUSD - Office of the Under-Secretary of Defence

P

PPO - Payload Operator workstations

POSIX – Portable Operating System Interface

PSM - Platform Specific Model

PWM - Pulse Width Modulation

Q

QoS – Quality of Service

QREN - Quadro de Referência Estratégica Nacional

R

R2C - Robot Command and Control

RA – Reference Architecture

RADAR - Radio Detection and Ranging

RAMP - Reference Advanced Model from Portugal

RC - Radio-Control

REX - Robotic Exercises

RF - Radio Frequency

ROS – Robot Operative System

RPAS - Remotely Piloted Aircraft System

RS - Recommended Standard

RTG - Research Task Groups

RVT - Remote Video Terminal

S

SAE - Society of Automotive Engineers

SAIL - Stanford Artificial Intelligence Laboratory

May referred to: SAR - Synthetic Aperture RADAR

SAR – Search and Rescue

SATCOM - Satellite Communications

SAS - Synthetic Aperture SONAR

SCI - Systems, Concepts, and Integration panel

SEC2 – Sistema Embebido de Comando e Controlo

SEP – Sistema Embebido de Payload

SIGINT - Signals Intelligence

SISO - Simulation Interoperability Standards Organization

SLAM - Simultaneous Localization and Mapping

SM - Specialist Meeting

SNA - Systems Network Architecture

SPURV - Self Propelled Underwater Research Vehicle

SQUID – Superconducting Quantum Interference Device

SS - Sub-Systems

STANAG - Standardization Agreement

SOA - Service Oriented Architecture

SOF - Special Operations Forces

SONAR - Sound Navigation and Ranging

SUNNY - smart unattended airborne sensor network for detection of vessels used for cross border crime and irregular entry

SUGV - Small UGV

SY - Symposia

T

TACAN - Tactical Air Navigation

TCP/IP - Transmission Control Protocol / Internet Protocol

TCS - Time Critical Strike

TIC - Thermal Imaging Camera

TLV – Type – Length – Value

TRL - Technology Readiness Level

U

UAM - Utility Acoustic Modem

UAV - Unmanned Aerial Vehicle

UCAP - Unmanned Rescue Capsule

UCS - UAV Control System

UDP - User Data Protocol

UGV - Unmanned Ground Vehicle

UML - Unified Modelling Language

UK - United Kingdom

U.S. - United States

USAR - Urban Search and Rescue

USB - Universal Serial Bus

USB IF - Universal Serial Bus Implementers Forum

USV - Unmanned Surface Vehicle

UUV - Unmanned Underwater Vehicle

UWWCG - Under Water Warfare Capability Group

UxS - Unmanned System

V

VDT - Vehicle Data Terminal

VHF - Very High Frequency

VSI - Vertical Speed Indicator

VSM – Vehicle Specific Module

W

WAN - Wide Area Network

WHOI - Woods Hole Oceanographic Institution

X

XDR - eXtensible Markup Language Data Reduced

XML – eXtensible Markup Language

XNS - Xerox Network System

1

Introduction

The aim of this chapter is to present the motivation for this work, the research question and the hypotheses generated by this question, the research method used, the contributions of this thesis and the structure that this thesis follows.

1.1. **Motivation**

Unmanned Systems (UxS) are changing the way military and civil operations are carried out. One of the reasons why development in this area is of great importance is that robotic systems do not require the logistical footprint that a human being requires to operate[1]. Such systems do not eat (but still need power) which reduces the logistic burden considerably. They do not rest or sleep (but need maintenance) and so will be operational 24 hours a day, reducing the need for multiple shifts. They don't suffer as much as humans from cold or heat (but their components have limits) and that's why they have fewer environmental restrictions. They don't have the training requirements of human beings, and the introduction of software upgrades can be easily implemented, which reduces the preparation time for each type of mission.

Each unmanned vehicle (UxV) is created with one main goal in mind and will have space and means to support sensors and equipments required to acaccomplish the mission[2]. With the increasing popularity of this kind of system, it's expected that there will also be an increase in demand, as well as further development. With the increasing demand, more people will want to thrive in this industry which in turn enables the development of more technology. Logistics, software, sensors and communication are just some of the areas that will be improved, and it's expected that their price will be lower, and their capabilities will be enhanced, both in the military and civilian applications[3].

Today each company develops its systems with their own structure, their own commands, and their own control station. In the end, this company may have a system that can operate very efficiently by itself, but in combined operations it will be difficult if not impossible to operate in conjunction with other vehicles. If each system was developed using a common reference model, interoperability would be much simpler.

The benefits of common reference model are obvious, but the main ones are:

- Interaction and cooperation between UxS will be easier to accomplish;
- A single ground station will be able to control more than one system, avoiding the "forest of computers" that is common when multiple vehicles are used[4];
- Software will be reusable between different systems, reducing costs and reducing the time necessary to develop the systems;
- Training (both for operation and for maintenance) will be easier because of common components;
- Prices can be lower because of scale economies (the same components are used in many different systems);
- Prices can be lower because there will be more manufacturers, and thus competition amongst them;
- The logistical chain will be simplified because there will be less different components;
- Finding a component can be easier because it will be used in many different systems and from different vendors;
- Maintenance will be easier since there are fewer systems;

- The learning curve for new competitors, universities, and even hobbyists will be smoother, thus fostering more development and innovation.

Standardization is thus essential in the development of robotic and automation systems. Lacking guidance and standardization in robotics may cause slower development, or lead to divergence of development, causing frustration not only for consumers but also manufacturers. Standardization not only facilitates commercialization and knowledge transfer, but also guides research and development activities towards more focused solutions[5].

If researchers and developers where more familiar with existing standards, data models, frameworks and protocols it would be easier to adopt them from the outset, thus simplifying later efforts to promote cooperation. This could lead to a qualitative change in the way we use UxS, by enabling the implementation of new concepts of operation for multi-UxS scenarios, and greater integrations of these systems[6].

Although many reference models have been proposed, most of these have a partial or specialized view of the general area of UxS, and none of them has a clear dominance of the market.

A single, all-encompassing and easy to use a reference model would be a great step ahead for robotics. However, this may be an illusion: one may ask if it is at all possible, and if it is, why hasn’t it appeared yet? This is the motivation for developing this thesis.

Throughout history, the military community has frequently been the driving force of technology and is at the root innovations that go from satellite navigation and airplanes to steam engines, optimization theory, and quality control. In fact, the defense of a nation requires these innovations and the military are the ones with money for that achievements. As a military, this is a big motivation to work in this area.

1.2. **Research Question**

Currently most military operations and many civilian activities too, require heterogeneous UxS[2]. In the foreseeable future the use of UxS will certainly increase significantly. There is also a tendency to increase the interactions between these systems, making them aware of each other, executing tasks that require cooperation (both by design and by self-organization), and finally implementing flock or swarm behaviors[7]. Although many standards have been proposed, most of these systems have their own command and reporting protocols, and consequently require their own ground control stations. This profusion of protocols makes it very difficult to implement cooperation between systems. Their operation and maintenance, in multiple vehicle environments, also poses an unnecessary burden due to lack of unified standards. A common solution is to develop wrappers and gateways from one system to another. This solution is generally sub-optimal in characteristics, and computationally inefficient.

A better solution would be to have either a single system of protocols designed from the start to work with each other, sharing a common view of what a UxS is and how it is organized. This is however an unrealistic expectation at this moment, given the development already achieved in many systems, and would probably suffer the same fate as other "all encompassing" approaches that have been attempted in technology, such as the Multics project of Bell Labs[8] in the 60s or the enforcement of Ada by the Department of Defense (DoD)[9] in the 80s. In fact, there may be no such "single system of protocols" that addresses all problems raised by UxS.

Therefore, the main research question chosen for this work is the following:

Is there a reference model that describes all components and issues concerning unmanned vehicles that are relevant to achieve interoperability of heterogenous groups of such vehicles, and a standard that following that reference model achieves that interoperability?

To answer the main research question of this thesis, the hypothesis proposed are:

H 1 -It is possible to achieve interoperability amongst heterogeneous unmanned vehicles if they all share a common reference model (which we propose) and use one of the existing communication methods to exchange messages.

1.3. **Research Method**

The research method used is based on the classical approach proposed by [10], and is composed of several steps[11]. The method is represented in Figure 1-1 and then explained in the following paragraphs.

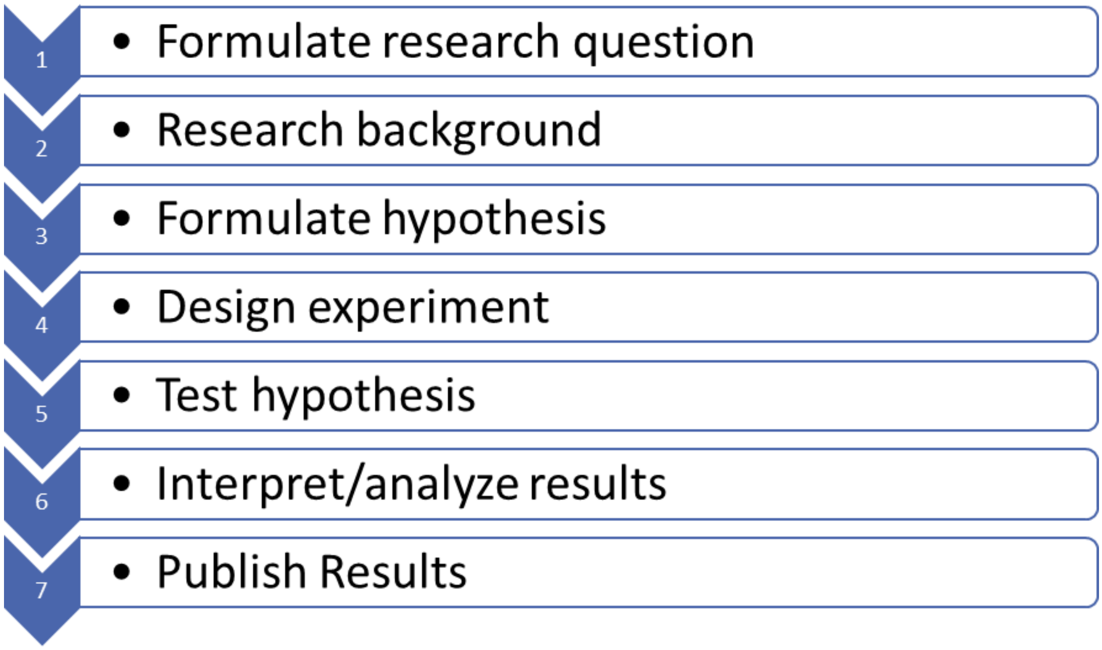

Figure 1-1 - Research Method

The first step is to *formulate the research question*. This sets the goal for all the thesis and all subsequent work should help to find an answer to this question.

Research background is the second step of the proposed research method. It is an essential step, as it implies extensive research about the subject and any other related work in the area. The scope of this research should be very broad, as it makes the next steps of the method easier.

Based on the information gathered on the step above, the third step is to *formulate hypothesis*. This is where the researcher proposes possible solutions to the problem introduced in the research question. However, the proof that they solve the problem is not developed in this step, and thus the solutions a still just hypothesis at this step.

As was mentioned before, the fourth step is to *design the experiment* and execute it. This is the first practical step of the research, and it is often implies designing a prototype, a system architecture, or other solutions or experiments.

The fifth step is to *test the hypothesis*, collecting data. In our case, it is important to evaluate the reference model proposed and to conduct tests, simulating in different scenarios which should be very close to reality, where we can assess the validity and applicability of the model.

After the realization of tests, the results should be *interpreted and analyzed* to validate the proposed solution. However, if the results are not satisfactory, it is possible to return to step 3, and formulate new hypothesis.

In order to validate the work done, it is important to *publish results*. The results should be presented to the scientific community to be shared and interpreted. This presentation is done through scientific papers, presented in scientific conferences or in journals.

1.4. **Contributions of the thesis**

The "holy grail" concerning interoperability issues would be finding a framework with protocols and tools that would encompass all possible needs and uses of autonomous vehicles. Reality is much harsher, because no single framework encompasses all these needs, and many different frameworks exist. Comparing these different frameworks, protocols, software tools, etc. is by itself a very challenging task, because there is no commonly adopted reference model where the different blocks can be "categorized" or "placed". Since this is a relatively new area in science and engineering, each community has developed its own conceptual model (even if not always explicitly shown), and comparisons are very difficult.

A similar problem arose in the 1970s, when computer networks started to be developed. Many problems had to be addressed, and each vendor or developer solved the issues using different conceptual models. Luckily, the *Comité Consultatif International Téléphonique et Télégraphique* (CCITT), later integrated with the International Telecommunication Union (ITU) promoted the development of a reference model named *Open System Interconnection* (OSI) model. This model defined different levels of communication and sorted the various problems to be solved in each layer. Although the protocols developed in direct accordance to this model had only moderate success (few people use X.25 or X.400 nowadays), it was extremely important to "organize the heads" of students, developers, and users, and led to an explosion of network solutions. Each person could say which problems or layers were addressed by their developments. We hope to provide a similar reference model to unite and make comparable the arsenal of tools, middleware, protocols, etc., that exist for autonomous vehicles. With a bit of "tongue in cheek" we named it *The RAMP* (from Reference Advanced Model from Portugal), since we hope it will be a ramp for a rapid and sustained development in this area.

Another example can be taken from computer science and the software industry, where frameworks such as the *Portable Operating System Interface* (POSIX) or computer graphics frameworks (as described e.g. in [12]) allowed a rapid development of their respective areas. Even the models used in single products, such as UNIX, shaped the view that an entire community has on a problem, and almost all modern operating systems can and are compared to UNIX.

Certain basic principles apply to all interoperability efforts. For example:

- Robustness to new developments. Interoperability should not curtail the development of new features. There should be a path that allows new features to appear without disrupting the existing systems. For example, many communication protocols provide this robustness by adhering to the Type-Length-Value (TLV) principle: when a new feature (type) is added, legacy systems can detect that the type is new, but will know how many bytes are used by this new feature (Length), and can "jump" over the extra information (Value);
- Clear separation between interface and implementation. Interoperability is mainly about defining interfaces between elements, and the focus must be on the interfaces and semantics, not on specific implementations of these interfaces. However, the existence of implementations is crucial to the success and widespread adoption, and specific implementation end up having a huge influence in how the interfaces develop;

- Independence from specific technologies and uses. Since technology develops rapidly, the interfaces must be independent of it and outlive a specific technology. Likewise, interoperability efforts that had one specific use in mind many times end up having a much broader application in new fields.

1.5. **Thesis Structure**

This thesis is organized in six major chapters. The first chapter presents the motivation that led to the choice of this topic; the research question and hypothesis; the method chosen to develop this research; the contributions of this thesis and its structure.

The second chapter introduces some concepts required to understand the thesis; overviews the history of unmanned vehicles; classifies unmanned vehicles in different categories; and presents some of the missions usually (or potentially) given to unmanned vehicles.

In chapter 3, we overview the most relevant interoperability building blocks used in unmanned vehicles and present some comparisons between them.

The fourth chapter explains the reference model we propose, that is applicable to all types UxVs.

Chapter 5 validates the reference model presented in chapter four by giving examples of UxV, used in research projects where we were involved, where the reference model fits perfectly. We also provide an example of a research project, using a fleet of heterogeneous UxV where not only the reference model fits, but where we actually used a common interoperability building block that allowed the desired interoperability.

Finally, in chapter 6, the answers to the research question and hypotheses are presented. We also discuss the integration of this work with other research activities, summarize the publications that resulted from our work on this thesis, and suggest future work.

2

Background

This Chapter explains important concepts used in this dissertation, provides a review of different types of unmanned vehicles, and a possible classification of unmanned vehicles. It also discusses the missions where they can be used.

2.1. Relevant Concepts

When it comes to the study of systems, or systems of systems[13], the concept of interoperability is paramount for the functioning of a system entity[14]. However, it is important to define some concepts that are fundamental to understand this work:

- *Standard*. A standard is a document that defines the characteristics, such as dimensions, safety or performance aspects, of a product, process or service[15]. A standard usually defines a set of rules and models that a system should have, or formats that they use. The implementation of standards can be facilitated by the use of common *frameworks* or functional structures between systems;
- *Framework*. According to the *Information Technology Standards and Organizations Glossary*,[16] a framework is “a real or conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful”. Usually, a framework has different layers of standards, and software that facilitates the use of those standards;

- *Architecture* can be defined as a conceptual model that describes the structure, organization, behaviours and components that compose the overall system[15];
- *Model*. The concept of model can be introduced at a system level, as being a representation of a certain object[17];
- *Data model* is an abstract way of defining how data is represented in system, aiming to conceptualize and structure the way information is represented and stored[18];
- *Middleware*. This is a concept that is used to refer to software that connects two different complex programs;
- *Structure* is the general arrangement, organization and disposition of the materials that make an object or a system that is more complex. When used in construction it is the arrangement of the fundamental elements of a building that allow it to stand. When used in UxS it is the arrangement of the components or "the skeleton" of the vehicle[15];
- *Format* may be the shape, size, general makeup, general plan of organization or arrangement of something. When referring to computers, it can also be a method of organizing data and how the information is encoded and stored;
- *Service* is the providing of activities or any other needs that are necessary for someone or something. In computer science a service might be a background activity, proving a function to other programs, that is required for the system to work[18];
- *Message*. In the context of UxS, a message is a block of information or data, organized according to a code, language, or predefined format, transmitted by an emitter to a receiver[19];
- A *package* may be a container, in which something is or may be packed, a group of information intended to be delivered to someone or a group of objects or activities[15];

- *Protocol*, is defined as a set of regulations that determine how the data should be transmitted in the network, usually specifying sets of messages to be used in the communication[15];
- *Communication method*. A communication method is a system that allows the exchange of information, or at least data, amongst system. In this thesis communication methods are mainly used to allow interactions between unmanned vehicles and their control stations, but communication methods are also used within the vehicles, within the ground control stations, and with external systems. In this context, communication methods can involve standards, data models, frameworks, and protocols;
- *Interoperability*. According to the Institute of Electrical and Electronics Engineers (IEEE)[15], interoperability is defined as the capability of a system to work with another without great limitations or additional effort from the user, which can only be achieved using standards. This is one conceptual view of interoperability, but interoperability is a broad concept which extends itself to a vast number of different areas. In health care, interoperability is known as "the ability of health information systems to work together within and across organizational boundaries to advance the effective delivery of healthcare"[20]. In telecommunications, it can be defined as the capability of providing and accepting services from different systems, enabling these services to work effectively together[21]. Within the North Atlantic Treaty Organization (NATO), it is defined as the ability of two or more nation's forces to train, exercise and execute effectively assigned missions and tasks[22]. When it comes to software, interoperability is the ability to communicate, transfer data and execute programs between different units which require little or no knowledge of the characteristics of the units to the user.

As can be seen, interoperability is very important for industry, for the armed forces and for many other organizations that want to have success in a common goal.

For the armed forces, interoperability is usually considered at four different levels (strategic, operational, tactical and technological):

- At a strategic level, interoperability is important as a coalition builder. It facilitates contributions between countries, for example, between NATO members. At this level, interoperability means sharing strategies, doctrines and definitions of force structures in such a way that the other nations understand them and can interact with them. The cost of not having interoperability at this level leads do a fallout amongst the coalition and problems at a political level;
- At operational and tactical levels interoperability requires that different forces share planning methodologies (and the plans themselves) and that training of different units is done using similar approaches, so that the forces interact seamlessly, know how their partners will react, and trust each other. Lack of interoperability at this level will result in not being able to deploy forces from different nations at the same time in the same scenario, thus missing out in possible synergies of different forces;
- At the technological level interoperability focuses on systems and interfaces. At this level, different systems must be able to exchange information and interact with other systems to achieve the desired effects. The rapid development of technology and its complexity has made interoperability at a technological level a major concern for NATO, that has a vast set of technical standards, named Standardization Agreements (STANAG), and recommends all nations to adhere to them so as to allow nations to work together and reduce costs [22].

2.2. Examples of organizations and processes involved in interoperability issues

There are organizations that are responsible for some areas of interoperability. These play an important role in allowing various stakeholders to interact with each other and have proved to be powerhouses for development. Their scope can vary from very specific local trade associations for a very narrow set of issues, to worldwide organizations involving governments or industry leaders. All play an important role and interoperability standards, models, or frameworks need their support to flourish. Some organization, such as International Organization for Standardization (ISO) or Deutsches Institut für Normung (DIN), are maintained by governments (or the United Nations themselves) and have as their sole purpose the regulation of standards. Others, such as IEEE or Association for Computing Machinery (ACM) are professional associations that saw the need to provide standards, guidelines, or recommendations within their professional areas. Organizations such as the Open Geospatial Consortium (OGC), Network Centric Operations Industry Consortium (NCOIC), Web Services Interoperability Organization, Universal Serial Bus Implementers Forum (USB IF), Bluetooth, etc, are associations of commercial companies, universities, or other stakeholders that formed those organizations to pursue common interests in very specific areas[23]. Specially important for the area of UxS we have NATO with its STANAGS, that address issues such as command and control protocols, ground control station layouts, data formats for images and video, etc.

We shall now overview some interoperability efforts that have had an important impact in their sectors.

2.2.1. **OSI**

In the beginning of the 80's computers were starting to be connected to each other, and various research projects (such as the DARPA ArpaNet projects) and companies (such as International Business Machines (IBM) with its Systems Network Architecture (SNA), or Xerox with Xerox Network System (XNS)) were starting to develop different networking systems. These systems were not interoperable, and in many cases, they were propriety systems with copywrite and patent protection. Networking was difficult to achieve, and difficult to address because there was no underlying model that all could agree on. Thus, even comparing the different systems was a difficult task.

At this stage, two standardization organizations, OSI and CCITT, tried to get together teams of experts to lay out the foundations of the emerging area of computer networks. In 1983 these two efforts merged and produced the" The Basic Reference Model for Open Systems Interconnection", commonly known as the OSI Model. This model defined 7 layers (Figure 2-1), specifying what problems should be addressed at each layer, and how these layers interacted. CCITT went on to define the "X" standards, that covered the different layers of the model. The model was thus "populated" with different standards. For example, X25 (that actually predates OSI) defines a packet switching system for level 3 and X400 a mail processing system that covers aspects from level 5 to level 7. More importantly, networking systems that were quite different from the 7-level model, such as Transmission Control Protocol / Internet Protocol (TCP/IP) that had developed out of ArpaNet, were mapped to the OSI level, and their features could easily be compared thanks to that mapping.

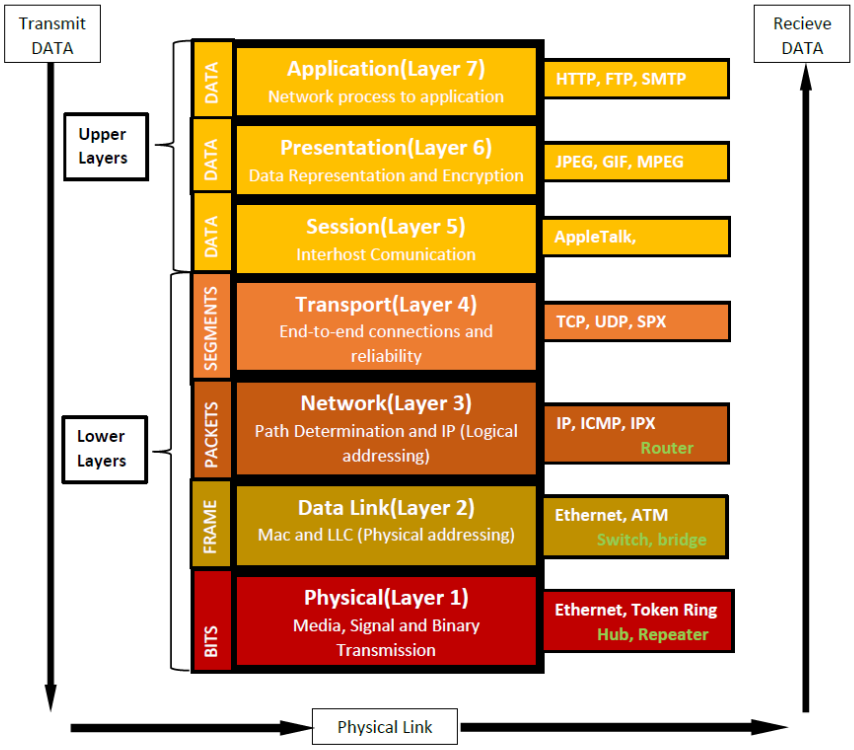

Figure 2-1 – OSI model

Thus, by providing both a reference architecture for computer communication, and a set of protocols to implement it, OSI contributed enormously to the development of computer networks, and the existence of the networked would we live in today. The fact that everyone knew the model and that there were protocols that everyone knew about (and many were free to use) let to a very rapid and widespread development. Even though the "x-standards" themselves, and in great part the spirit of OSI have not been followed in present day internet (e.g. comments on OSI vs TCP/IP in [24]), they did provide the common ground for all protocols.

2.2.2. ITIL

The Information Technology Infrastructure Library (ITIL) is an important set of rules that should be considered for the operation and management of technological systems. It was developed and introduced in 1990, and consists of a set of documents and libraries with the objective of promoting the management of information and technological systems, in order to deliver to the client, the best product possible[25]. This library enables organizations to use common practices to identify, plan, deliver, improve and support IT services. As an example, it defines the IT Service Management (ITSM), as is represented in Figure 2-2. In this cycle, five phases are used: strategy, design, transition, operation and improvement.

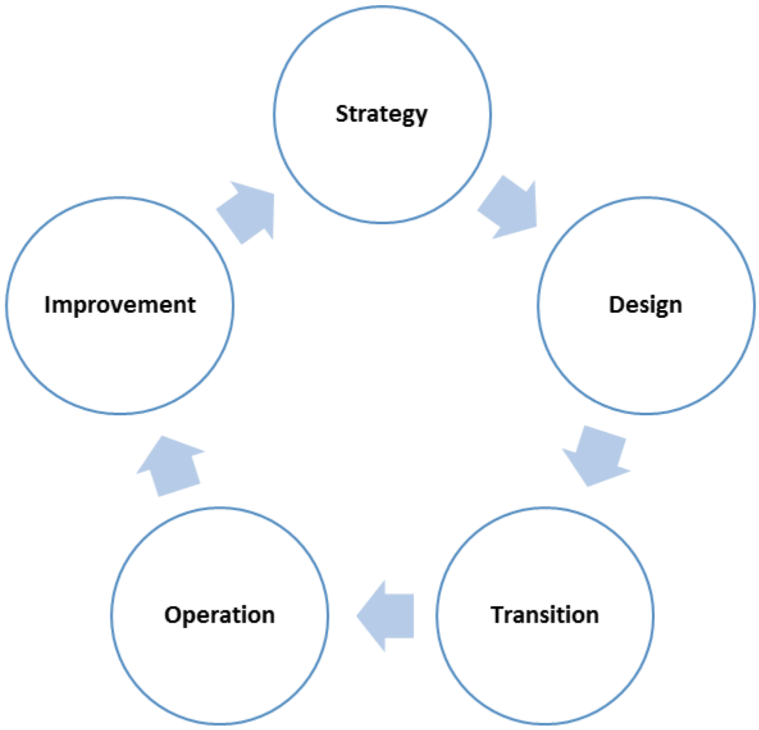

Figure 2-2 - IT Service Management Lifecycle

Although originally planned to support United Kingdom (UK) government efforts to stream-line IT services, ITIL has become a worldwide reference for most IT service providers, and has contributed significantly to the development of this sector (e.g.[25]).

2.2.3. **OODA Loop**

Although not formally a standard, but in practice a useful and wide-spread conceptual framework, especially in the UxS community, we have the "OODA Loop". This stands for *Observe, Orient, Decide and Act* (OODA), and is a cycle of steps for a decision process. This process can be seen in Figure 2-3. It was developed by United States Air Force Colonel John Boyd (e.g.[26]). This loop was originally developed for strategy in military operations and combat scenarios, but it can be adapted to suit almost any decision process. Nowadays, this model is dominant in Command and Control (C2).

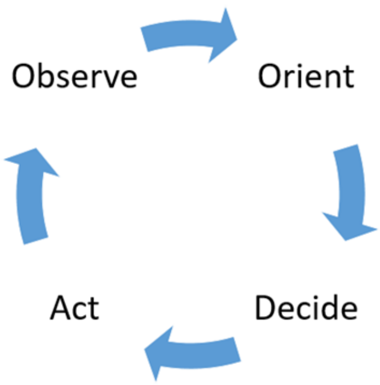

Figure 2-3 - OODA Loop

As can be seen in Figure 2-3, the ODDA cycle is a loop, where *observe* is the act of gathering information about the environment. In the case of UxS's, this can be the acquisition of targets or images. The next phase is *orient*. This refers to pointing the system to a certain function. This phase filters the observed information, through the experience, culture, ability to analyse and synthetize. This can be applied to UxS when filtering the information that was gathered, so as to only keep what is important. The *decide* activity is where the decision process is actually done. This is done based on the information that was gathered and filtered. Decision leads to the final phase, *act*. Act stage is doing the task that was decided previously. After this, another observation is done, and the loop continues (e.g.[27]).

By providing a way of looking at the decision problem, even though no written standards impose anything in particular, a standard such as this does in fact enhance interoperability.

2.3. **Relevant Milestones for Unmanned Systems**

Unmanned systems have been used throughout most of recorded history, but we are interested only in those that have significant autonomy. In the following sections we shall review some of the important landmarks in their development. We shall start with the air vehicles, since these have had the greatest impact, and will then go on to maritime (surface), ground vehicles and underwater.

2.3.1. **Unmanned Aerial Vehicles (UAV)**

Nowadays, UAVs are becoming common equipment. It's a technology that has been evolving from the military to the civilian world. The precursors of this technology date back to the nineteenth century, with the use of kites and balloons to spread information and to deliver air strikes. One of the first uses of UAVs was in 1806, when an officer from the British Navy, Thomas Cochrane launched kites using a frigate's guns to deploy leaflets over France[28]. Fifty-three years later, Austrians deployed balloons (Figure 2-4), armed with bombs, over Venice. A few years later, balloons used as unmanned aerial bombers started to be patented by Charles Perley.

Figure 2-4 -Air raid using balloons

Source: [29]

In World War I, several attempts were made to use unmanned airplanes. The most successful was probably "The Aerial Target" (Figure 2-5), developed by the British Royal Aircraft Establishment in 1916. This airplane was radio-controlled, launched from a truck, and used for target practice[30].

Figure 2-5 – The Aerial Target.

Source: [31]

In following year, Elmer Sperry and Peter Hewitt (that had been working on the concept of an "aerial torpedo" for some time) built the "Hewitt-Sperry Automatic Airplane" (The Flying Bomb) (Figure 2-6), which managed to fly 80 Km with 136 Kg bomb, with the aid of a launching platform.

Figure 2-6 - The Hewitt-Sperry Automatic Airplane.

Source: [32]

This success led to the construction of rail-launched Kettering Aerial Torpedo "Bug", for the U.S. Army, by the Dayton-Wright Airplane Company, in 1918[33]. In the European front, Germany also developed a similar project, "The Siemens Torpedo Glider", which could be dropped by an airplane and remote controlled via radio.

During World War II, in 1943, the German remote guided missile "Fritz X" (Figure 2-7), successfully sank an Italian vessel proving that Germany was reaching new technological ground with the advancements on the remote-controlled vehicles/missiles.

Figure 2-7 Fritz X.

Source: [34]

These developments led to creation of the "V-1 Flying bombs" (1944) (Figure 2-8) or the Vergeltungswaffen, a jet airplane first used in 1944 against Britain, resulting in great casualties[35]. Contrary to the Fritz X and other pre-existing UAVs, the V-1 was not remotely controlled by radio, but was indeed autonomous, in the sense that after its launch it had no control link.

Figure 2-8 - V-1 the German Flying Bomb

Source: [36]

The United States (U.S.) also contributed for the development of the UAVs technology by responding to German menace with their experiments, namely in operation APHRODITE and Project Anvil, in 1944 (Figure 2-9). This consisted in a two-man crew that operated the BQ-7 aircraft until a certain range of its target. When inside that range, the pilots would abandon the aircraft leaving it to be remotely controlled from a B-17. World War II marked a substantial evolution of the development of the unmanned technology. Despite some shortcomings the technology was promising, which resulted in further development during the Cold War years[37].

Figure 2-9 - Operation APHRODITE and Project Anvil

Source: [38]

During the Cold War, unmanned systems were used to perform missions of reconnaissance and surveillance. This was possible because of the progress of propulsion and guidance sub systems. The beginning of the 1950's (1951) was marked by the creation of the jet-propelled subsonic Ryan "Firebee" UAV (Figure 2-10). These UAVs were originally used for target practice, but later on modified for reconnaissance tasks and renamed to "Firefly" (The Lightning Bug).

Figure 2-10 - Drone control aircraft carrying two BQM-34S Firebee target drones

Source: [39]

By the end of the decade, the U.S. Navy had developed the DASH (QH-50) (1959) (Figure 2-11), an unmanned helicopter used for anti-submarine warfare. Remotely controlled from a ship, manned aircraft, or ground vehicle, this UAV carried homing torpedoes and depth charges, and was also capable of deploying sonobuoys and flares[33],[37]. However, the high rate of accidents with the DASH and their poor reliability stopped it from being used operationally throughout the fleet.

Figure 2-11 - The QH-50C DASH UAV being recovered aboard a ship

Source: [40]

In the Vietnam War the use of UAVs increased particularly for reconnaissance missions. New versions of the Lightning Bug capable of carrying larger payloads and taking photographs from various altitudes were used for these purposes. In 1965 the first UAV with stealth characteristics, the "Compass Arrow", producing a low heat and RADAR signature, was created by Ryan Aeronautical (now Northrop Grumman). The end of the Vietnam War and the focus on new cruise missile systems and long-range bombers affected the development of UAVs, effectively stalling it, until the next ensuing conflict[33].

The creation of the "Pioneer" (Figure 2-12), by the Israeli Aircraft Industries, established a new standard for UAVs. Israel had used a previous version of the airplane during the conflict in Beqaa Valley in 1982, and a lot of experience was gained on the operation of UAVs. This got the attention of the U.S. DoD, and the Pioneer started being developed in 1986 for the U.S. forces. It was a small propeller aircraft that became famous for being used by the U.S. during the First Persian Gulf War. It was used to deliver air-raids with effective results and without risking human lives. The "Pioneer" has been commonly employed in operations in Bosnia, Haiti and Somalia, as well as in the War on Terror[33],[37].

Figure 2-12 The Pioneer UAV

Source: [41]

In 1994, following the success of the Pioneer, the "Predator" was developed (Figure 2-13). It is a high-endurance and high-altitude UAV, capable of employing fire power and equipped with a Synthetic Aperture RADAR (SAR) for better discrimination and terrain imaging. These features were used for Intelligence, Surveillance and Reconnaissance (ISR) missions. After further development, this UAV can now carry out more offensive missions due to the capability of carrying a laser designator and missiles. The Predator set the stage for the development of the ISR capable “Global Hawk”, “Hunter” and “Shadow”, UAVs with an extended autonomy and capable of carrying a heavier and more technological advanced payload[37],[33].

Figure 2-13 - The Predator UAV

Source: [42]

Recently, small and micro UAVs have been developed, for both civilian and military purposes. The usage of small UAVs like the “Raven”, “Dragon Eye” and the “Boeing ScanEagle” (Figure 2-14) proved to be very effective. UAVs are now employed in ISR missions, discreetly providing terrain images, with mission endurance up to 15 hours.

Figure 2-14 - Launch of a Boeing ScanEagle

Source: [43]

An even smaller class of UAV (micro-UAV), like the "Wasp" (Figure 2-15), has emerged recently. These handheld and hand launched systems have shown versatility, proving to be valuable in different types of warfare, ISR missions, search and rescue, agriculture, law enforcement, meteorological services[37],[33].

Figure 2-15 - The AeroVironment Wasp.

Courtesy AeroVironment, Inc.

Figure 2-16 summarizes the evolution of UAVs, since their inception.

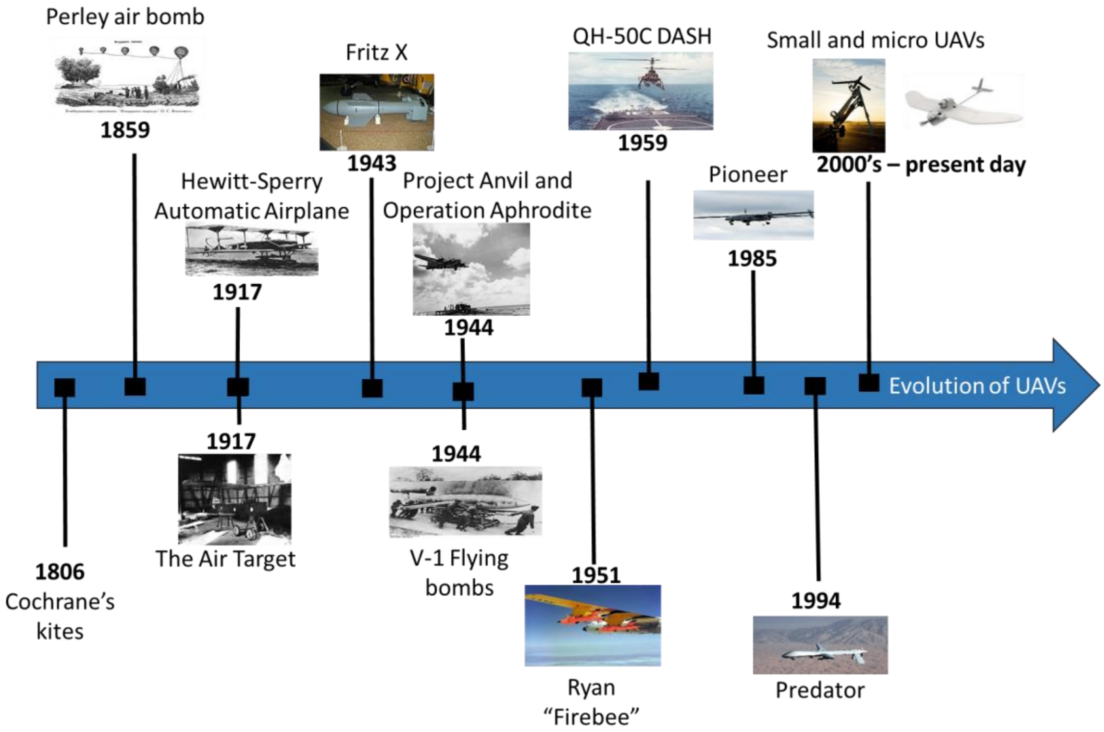

Figure 2-16 - Evolution of UAVs

2.3.2. Unmanned Surface Vehicles (USVs)

Historically, the most well-known Unmanned Surface Vehicles are probably the fireboats used by the British against the Invincible Armada in 1588[44]. This was at the time a common practice, used by many countries. These fireships were rigged by a crew that then abandoned ship and left it to run with the wind. Some later version had in effect mechanical auto-pilots that could, to a certain extent, keep a course even if there were slight changes with the wind.

The first remotely controlled Surface Vehicles are due to Nikola Tesla's work on radio-control. In 1898, he was able to remotely control a small motorboat, the "Teleautomatons" (Figure 2-17) with an electromechanical radio receiver and actuators that controlled the steering system[45].

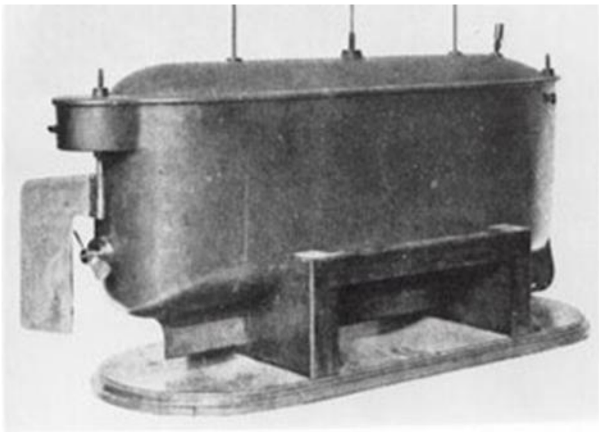

Figure 2-17 - The Tesla "Teleautomaton"

Source: [45]

Remotely controlled surface vehicles became very useful once they were used as weapons. In 1909, Gustave Gabet ran tests (in the Seine, Paris) on his *Torpille Radio-Automatique* (Figure 2-18), a steerable floating remotely controlled torpedo[46].

Figure 2-18 - Gabet and his "Torpille Radio-Automatique"

Source: [29]

After World War II, the usage of unmanned surface vehicles was mainly for testing of nuclear weapons. By 1946 Apex Drone Boats (Figure 2-19) were commissioned to collect post detonation water samples filled with radioactive compounds, to study the effects it would have upon vessels[47].

Figure 2-19 - Decontamination of Navy Apex Drone Boats

Source: [47]

During the Vietnam War, USVs were used as remote minesweepers, for example at Nha Be[48]. This later led to very successful minesweeping systems such as the European Tripartite System[49] .

The Massachusetts Institute of Technology (MIT) Sea Grant college program of 1993 generated a substantial leap forward in the development of USVs. The first prototype was ARTEMIS (Figure 2-20), a USV designed to test autonomous navigation and control systems, and later used to collect bathymetry data [50].

Figure 2-20 - The MIT ARTEMIS

Source: [51]

The ARTEMIS project gave birth to the Autonomous Coastal Exploration System (ACES), followed by the "Autocat" (2000) (Figure 2-21). These USVs were equipped with hydrographic survey sensors that resulted in the improvement of the quality of the surveys[50].

Figure 2-21 - The Autocat

Source: [50]**.**

With the turn of the century, there was a boost in the appearance of USVs powered by "renewable energy". These vehicles can harvest energy from the environment, either using wind, solar radiation, or wave energy. One of these USVs is Barlavento (Figure 2-22) developed at The Portuguese Naval Academy[52], that uses rigid sails to capture wind energy[52].

Figure 2-22 – Barlavento

Photographed during the "Robotic Sailing Regatta" of the WRSC, September 2016, Viana do Castelo

Liquid Robotics's small Wave Glider uses the waves as power source (Figure 2-23) have shown very promising results, indicating its use for possible surveillance and scientific missions[50]. While the waves produce the motion of the vehicle, it also has solar panels on top, so as to produce the electricity it needs for on-board electronics. The Portuguese navy first tested one such system in 2013 during the REP exercise (originally “Rapid Environmental Picture”, and now “Recognized Environmental Picture”) and since then they have been regularly used in this exercise.

Figure 2-23 - Wave Glider USV by Liquid Robotics

Image courtesy Liquid Robotics, a Boeing Company

Figure 2-24 summarizes the evolution of USVs, since their inception

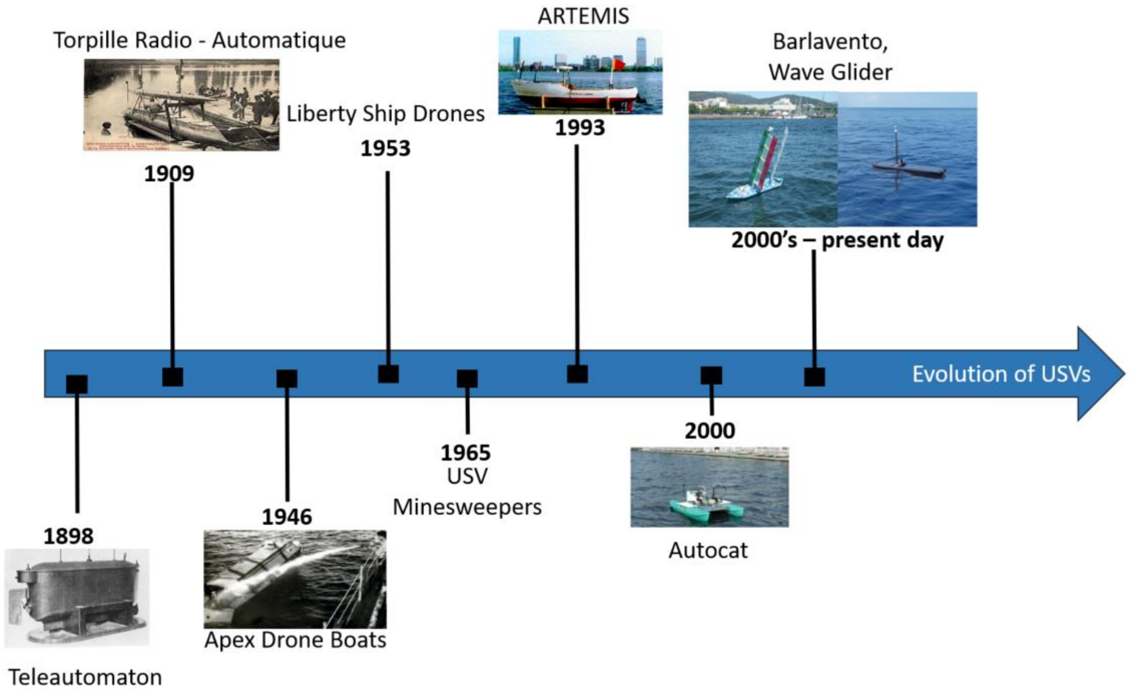

Figure 2-24 - Evolution of USVs

2.3.3. Unmanned Ground Vehicles (UGVs)

The Wickersham Land Torpedo (Figure 2-25) was probably the first UGV. Created by Elmer E. Wikersham, it was patented in 1928, and was built to deliver explosives to the target. Although it only remained a prototype, it set the pace to future creations[45].

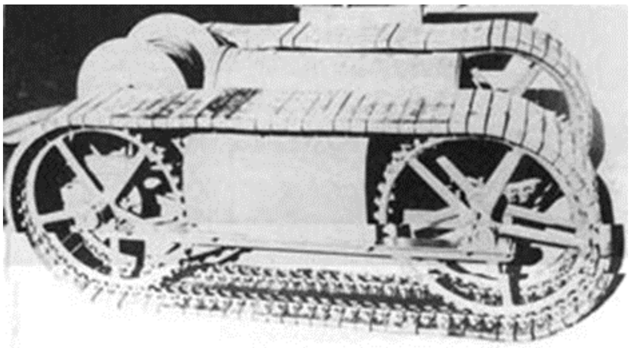

Figure 2-25 - Prototype of the Wickersham Land Torpedo

Source: [45]

In the 1930s, the Russians developed the first UGV used in conflicts, the "Teletank" (1930) (Figure 2-26), that was a full-sized tank.

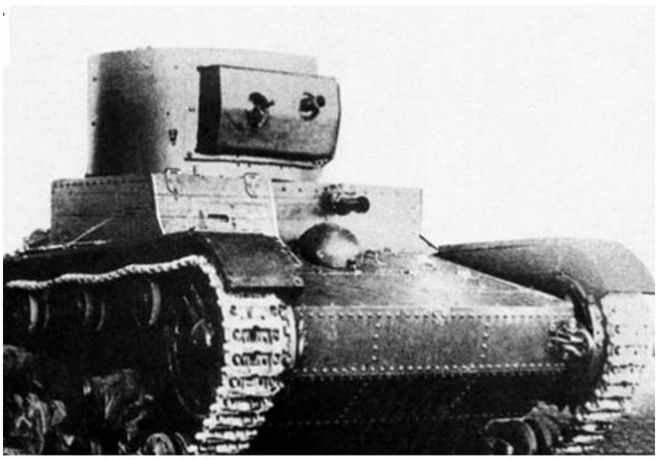

Figure 2-26 – Teletank

Source: [53]

These tanks were used during the Winter War, were controlled remotely across the distance of almost one mile, and could carry machine guns, flamethrowers and bombs. Based on a French design, the Germans also developed this technology by creating the "Goliath" (1942) (Figure 2-27), a remotely controlled small tank that was used to reach the enemy and to be remotely exploded, serving as a portable bomb during World War II[54].

Figure 2-27 – Goliath

Source: [53]

One of the greatest innovations of UGV technology occurred with the creation of "Shakey" (Figure 2-28) in the research department of Stanford University by the Defense Advanced Research Projects Agency (DARPA), in 1969. This was a wheeled mobile robot that used a video camera, a radio link to the controlling computer, a laser range finder and a blocks-world image-reading algorithm to perceive its surroundings[55],[56].

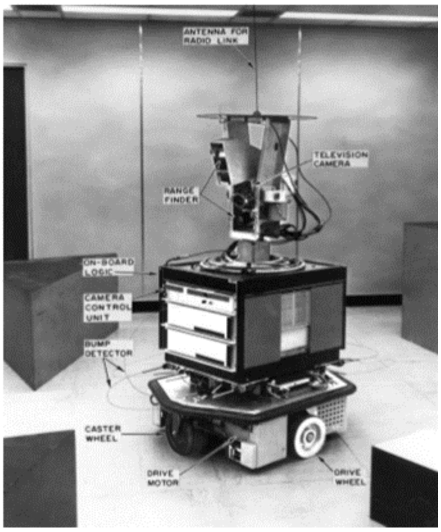

Figure 2-28 - Composition of "Shakey"

Source: [56]

Along with Shakey, in 1971 the "Stanford Cart" (Figure 2-29) was developed, showing progress by being capable of performing autonomous outdoor movement at a steadier pace. It could follow an unbroken line on a road for about 15 meters independently. These developments marked the beginning of the implementation of autonomous navigation systems based on artificial vison[55].

Figure 2-29 - The Stanford Cart displaying autonomous movement

Source: [57]

As for remotely controlled vehicles, progress was also made by the development of the "Wheelbarrow" (1972) (Figure 2-30), by the British Army. The first Explosive Ordinance Disposal UGV, used in Northern Ireland in response to casualties caused by the Irish Republican Army[54],[58].

Figure 2-30 - The Wheelbarrow used as a bomb disposal tool

Source: [58]

Along the 1980's, DARPA developed the "Autonomous Land Vehicle" (1986) (ALV) (Figure 2-31), that was capable of autonomous on-road and off-road driving and was a self-contained system. It was an eight-wheel vehicle with an inertial land navigation system, ultrasonic sensors, a doppler RADAR, color video camera and a custom laser scanner used for perception. By the end of the decade, it performed off-road programed routes, being capable of obstacle avoidance [55].

Figure 2-31 - The DARPA's Autonomous Land Vehicle

Source: [59]

With the beginning of the 1990's, in 1992, DARPA initiated the DEMO II program (Figure 2-32) that focused on point-to-point cross-country routes that could simulate those of military scout missions. The vehicles were equipped with black and white video cameras, laser detection and ranging systems, forward looking infra-red systems and passive sensors. These were more robust vehicles with better navigation systems than their predecessors. By the end of the decade, the project showed improvements in road following technology and in obstacle avoidance, setting the standard for the new century[55].

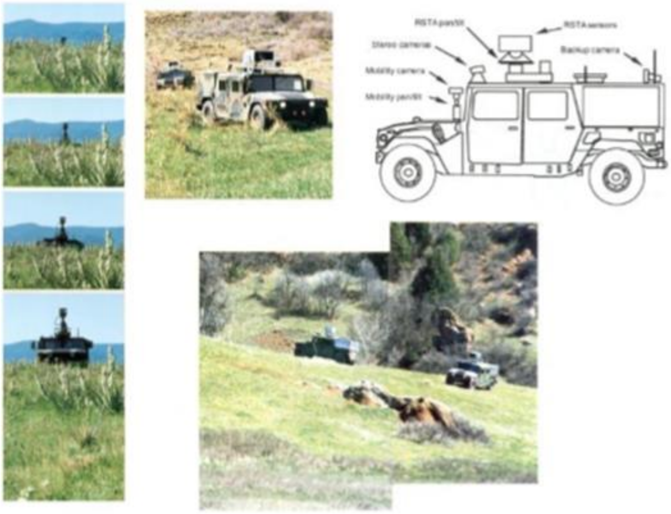

Figure 2-32 - The DEMO II vehicle and environment

Source: [55]

The current century has resulted in great improvements in UGVs, such as the Mobile Detection Assessment and Response System (MDARS). It uses arrays of sensors to detect obstacles and detect intruders. Another example is the Ironclad (Figure 2-33), whose purpose is to rescue injured soldiers during conflict.

Figure 2-33 – Ironclad

In the civilian realm, UGVs are gaining a lot of popularity, with innovations such as the Google Self-Driving Vehicles, Self-driving cabs, and self-driving trucks[60].

Figure 2-34 summarizes the evolution of UGVs, since their inception

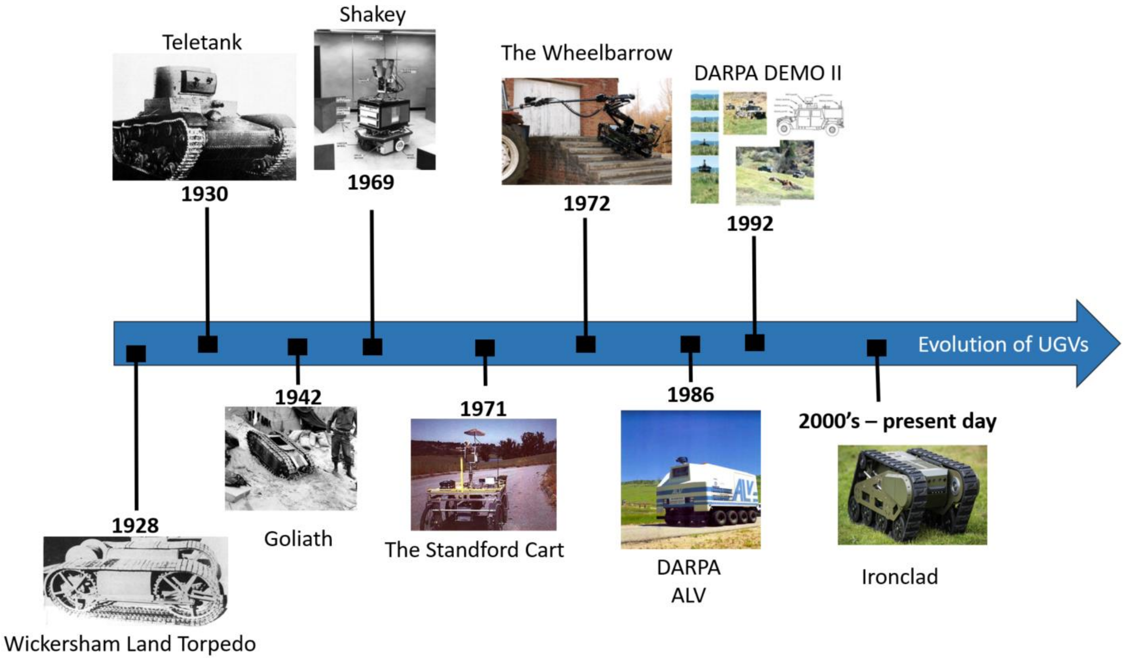

Figure 2-34 - Evolution of UGVs

2.3.4. **Unmanned Underwater Vehicles (UUVs)**

In 1957, what was probably the first UUV was conceived by Stan Murphy, Bob Francois and Terry Ewart, in the Applied Physics Laboratory of the University of Washington. This UUV was initially intended to collect oceanographic data in certain regions and under ice. This project triggered the development of "The Self Propelled Underwater Research Vehicle(s)" (SPURV) (Figure 2-35), a project that unfolded during the sixties and continued until the mid-seventies. The first version, the SPURV I[61], was controlled from the surface, via acoustic signals, and it could navigate at constant pressure, reach depths of up to 3 km, and had an autonomy of almost six hours. It was used to collect data at isobaric lines in order get information for creating models for wave studies. Further versions of the SPURV were used to study submarine wakes and acoustic transmission, along other oceanographic studies[62].

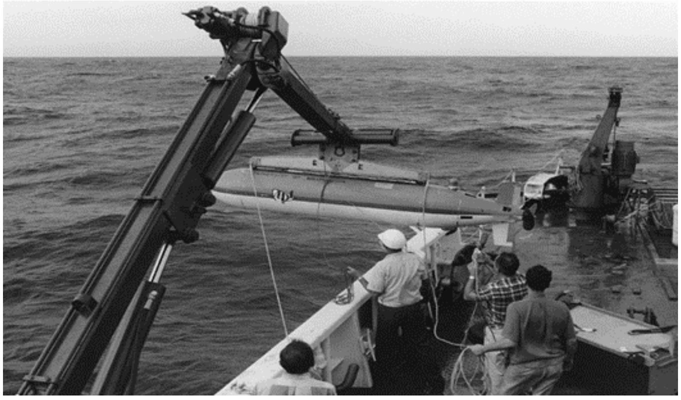

Figure 2-35 - Deployment of the SPURV I

Source: [63]

In 1980, the IFREMER's Épaulard (Figure 2-36) was operational, being the first UUV to support deep ocean photography and bathymetric surveys[64].

Figure 2-36 – Épaulard

Source: [65]

The Advanced Unmanned Search System (AUSS) began to be developed in 1973 by the Naval Ocean System Center, and it was first deployed in 1983 with the objective of transmitting video images via an acoustic communication system, providing the capability of underwater reconnaissance (Figure 2-37).

Figure 2-37 – AUSS

Source: [66]

The 90's the production of UUV prototypes increased with the creation of Autonomous Benthic Explorer (ABE) (1991) by the Woods Hole Oceanographic Institution (WHOI) (Figure 2-38), the "Odyssey" (1992) vehicles by the MIT Sea Grant AUV lab (Figure 2-39), the International Submarines Engineering's "Theseus" (1995) (Figure 2-40), the WHOI's "REMUS" (Figure 2-41) and the Southampton Oceanography Center's Autosub.

Figure 2-38 – ABE

Source: [67]

Figure 2-39 – Odyssey

Source: [68]

Figure 2-40 – Theseus

Courtesy International Submarine Engineering Ltd.

Figure 2-41 - The WHOI's REMUS

These UUVs performed mainly scientific missions with the intent of operating under ice, performing near bottom surveys in rough terrain, laying fiber optic cables under water and ice and ocean monitoring[69],[70],[71],[72].

With the new century, UUVs began to be commercialized to the public. The first enterprise to do so was C&C Technologies of Lafayette, by selling Hugin 3000 UUV (Figure 2-42), for charter. This is a rather large system, used mainly by the oil industry and other large corporations. For less demanding uses, smaller UUV have been developed, with the capability of carrying various types of payloads, and equipped with cameras which allow recording of video and still images[73]. Several Portuguese companies have been involved in this area, such as OceanScan with their LAUV (Light Autonomous Underwater Vehicle) and INESC-TEC with its Mares and Tri-Mares system.

Figure 2-42 - Hugin 3000

Photographed during a research project meeting in which CINAV participated in October 2017.

Figure 2-43 summarizes the evolution of UUVs, since their inception.

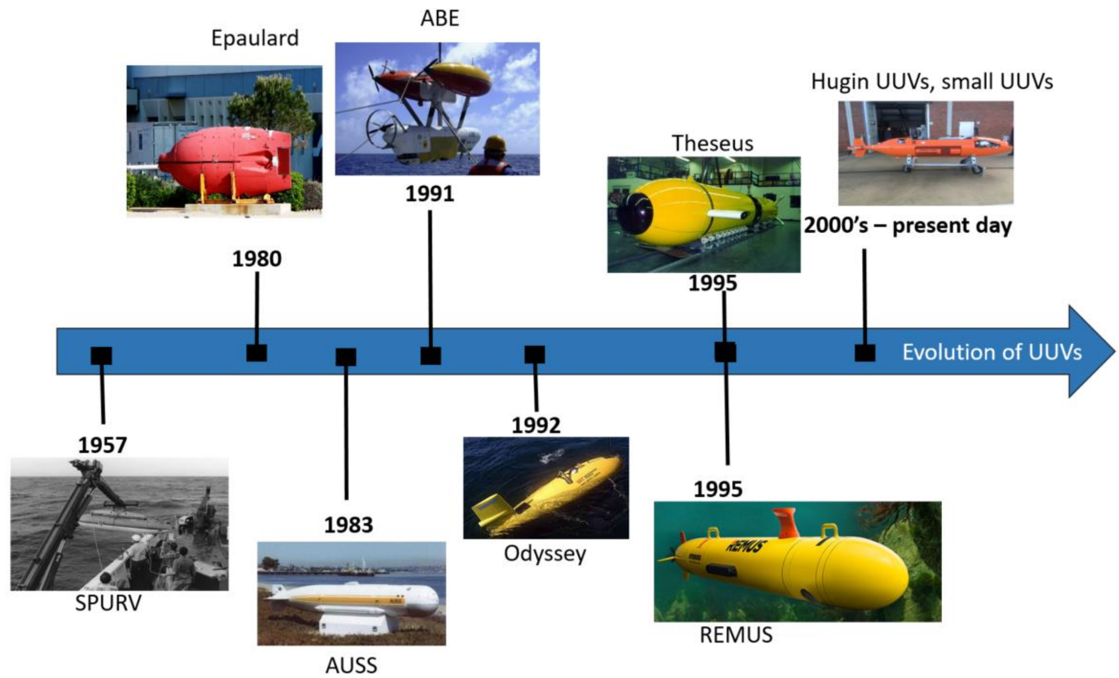

Figure 2-43 - Evolution of UUVs

Table 2-1 represents the evolution of unmanned vehicles:

Table 2-1 - Chronological evolution of unmanned systems.

	XIX Century	1910s – 1920s	1930s – 1940s	1950s – 1970s	1980s – 1990s	2000s
UAVs	Cochrane Kites Perley Air Bomb	The Aerial Target H.S. Automatic Airplane Bug	Fritz X German V-1 Op. Aphrodite	BQM-34S Firebee Dash QH-50C	Pioneer Predator	Small UAVs
USVs	Teleautomaton Tropille Radio- Automatique		Apex Drone Boats	USV Minesweepers	ARTEMIS Autocat	Solar USV Wave Gliders USV
UGVs		Wickersham Land Torpedo	Teletank Gollath	Shakey Standford Cart Wheelbarrow	ALV DEMO II	MDARS Ironclad
UUVs				SPURV	Épaulard AUSS ABE Odyssey Theseus REMUS	Hugin UUVs Small UUVs

2.4. **Classification of Unmanned Systems**

The name "UxV" covers all vehicles that do not have a person aboard with capability to control the system[74]. Thus, the vehicle must have other means of control, either being fully autonomous, semi-autonomous (programmed to follow pre-defined waypoints), or teleoperated, *i.e.* remotely controlled[75]. In this thesis we chose to use the names currently in use with most NATO bodies, that focus on the common fact that all these vehicles are unmanned in the sense that they do not have a human aboard to control (at least in part) the vehicle. In certain fora the names used focus on other aspects. In most European organizations the term Remotely Piloted Aircraft System (RPAS) is used to stress that there is always someone responsible (a Pilot), even though he may be physically outside the plane. In popular parlance, the term Drone is frequently used, but this term usually implies that the system is not intelligent and is simply a slave (and thus it gained a bad name in most military organizations). Other communities, such as the traditional underwater vehicle community, prefer to use terms such as "Autonomous Underwater Vehicle" (AUV) to stress that the systems must make choices by themselves during the mission. However, there have been objections to the liberal use of the term autonomy, since it is usually pre-programmed in some way (see the discussions on remote control/remote supervision/consented autonomy/full autonomy). To avoid all these pitfalls, we choose to always use the term "Unmanned".

These vehicles are usually classified according to where they operate: in the air (UAV), on the surface of water (USV), on the ground (UGV) or under the surface of water (UUV) (Figure 2-44).

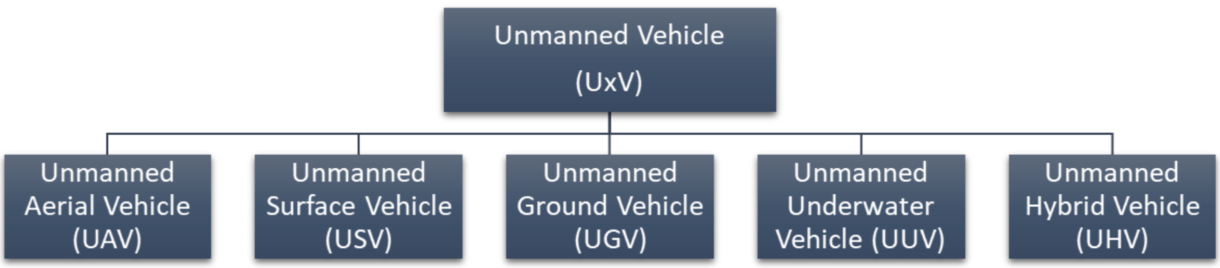

Figure 2-44 - Unmanned Systems divided into categories

They have similar broad objectives, and have similar building blocks or structure, but each one adjusts to its own operating environment and its specific components. In many applications we need to have different types of UxV cooperating to perform a task. Thus, interoperability between these vehicles is currently a hot topic all over the world and will continue to be a major issue for a reasonable future[76].

2.4.1. Unmanned Aerial Vehicles (UAVs)

Vehicles that operate in the air are designated as UAVs. These platforms can be controlled by electronic equipment present on the vehicle, coordinated by the goals of its mission or on a Ground Control Station (GCS), manually or by waypoints or other high-level references[77],[78]. UAVs can be classified by their operating altitude (Low, Medium, High), type of wing (rotary or fixed)[79], or by their weight[80] (Table 2-2). This latter classification can be applied to all UxV, since all have weight, and classifying according to it makes sense in all environments.

Table 2-2 - Classification of UAVs, USVs, UGVs, UUVs.

Class	Weight (kg)
Class I - Light	< 150
Class II - Medium	150 – 600
Class III - Heavy	≥600

Source: [81], [82]

Light UAVs are characterized by having small dimensions and having the capability of being operated by a small crew or even one person, and usually they can be launched by simple systems like hand launching or catapults[79],[82]. The typical flight time varies between 60 minutes and 15 hours and they can carry payloads to a maximum of about 50 kg. The most common on-board sensors are video cameras, Infra-Red (IR) cameras, nanoSARs and other equipment necessary to provide ISTAR. Some examples are the Boeing Insitu ScanEagle, the Wingo Ogassa, Ouranos (Figure 2-45), the UX-Spyro Quadcopter (Figure 2-46) and the Silent Falcon.

Figure 2-45 - Ouranos

Photographed during the final demonstration of research project SUNNY in April 2018, São Jacinto, Portugal

Figure 2-46 - UX-Spyro Quadcopter

Photographed during Robotic Exercise July 2016, at Lisbon's Naval Base

Medium UAVs are typically aircrafts that require several operators to deploy and keep in air operations, as have a more complex launching mechanism or runway for take-off. The typical flight time varies between 2 to 9 hours and they carry a maximum payload of about 150 kg. The most common on-board sensors are the same as those carried by Light UAVs, adding heavier and more capable electro-optical video systems, small RADAR systems, electronic warfare equipment and Signals Intelligence (SIGINT) equipment[79],[82]. Some examples are the AAI RQ-2 Pioneer, the AAI Shadow and the SKELDAR V-200 Maritime (Figure 2-47).

Figure 2-47 - SKELDAR V-200 Maritime

Courtesy of SKELDAR

Heavy UAVs are typically large, highly capable aircraft that require operation facilities similar to manned aircrafts, like runways, hangars and ATC. However, they provide more capabilities than their smaller counterparts. The typical flight time is more than 18 hours and they can carry payloads with more than 200 kg. The most common on-board sensors, besides the ones used in the other classes, are various types of RADARs, atmospheric and environmental sensors, wide area surveillance sensors, and specific mission packages such as weapon systems. Some examples are the General Atomics Predator/Guardian, the Northrup Grumman Global Hawk (Figure 2-48) and the Northrup Grumman Fire Scout (MQ-8C) (Figure 2-49).

Figure 2-48 - The Northup Grumman Global Hawk

Source: [83]

Figure 2-49 - The Northrup Grumman Fire Scout (MQ-8B)

Source: [84]

2.4.2. Unmanned Surface Vehicles (USVs)

USVs, are vehicles that operate on the water surface[85]. The main concern with these platforms is that, using only the available sensors, they have to manoeuvre at sea, avoiding collisions and calculating the best route to a goal based on weather and sea data/conditions[86]. Sometimes these platforms are used to deploy or control UUVs and UAVs [50]. USV's can be classified by their length (X-Class, E-Class, F-Class)[87]or by their weight (Table 2-2)

Light USVs are small, lightweight, portable vehicles that mostly can be carried and deployed by a small team of individuals and usually don't require complex systems to operate and to deploy[88]. They are mostly tele-operated, with some models being capable of semi-autonomous movement, and are very dependent of the sea and weather conditions. Their typical endurance is of around 3 hours (if they do not harvest energy from the environment) with a payload capacity of a maximum of about 70 kg. Most common on-board sensors are video cameras, weather instruments, Sound Navigation and Ranging (SONAR), hydrophones and other acoustic sensors, GPS and radio-locators[89]. Some examples are the USV I-1650, the Catarob, SailingFAST (Figure 2-50) and the GeoSwath 4R USV.

Figure 2-50 – Autonomous SailingFAST

Faculdade de Engenharia da Universidade do Porto (FEUP) Autonomous SailingFAST photographed during a test with the Portuguese Naval Academy off Cape Espichel, in 2011.

Medium USVs are typically larger vehicles that require more complex deployment systems like cranes and/or a pier. They can be tele-operated or capable of following predefined routes, just like UAVs. The most common on-board sensors besides those used in Light USV, are thermal cameras, communications relay devices, and Chemical, Biological, Radiological and Nuclear (CBRN) detection equipment[90]. Some examples are the ROAZ II (Figure 2-51) and the ASV Ltd. C-Cat 3 Small Multi-Purpose Work ASV.

Figure 2-51 - ROAZ II

Photographed during Robotic Exercise July 2014, at Lisbon

Heavy USVs are typically larger, more capable vehicles that may utilize modules designed to conduct specific tasks. Larger vehicles are more autonomous, with some smaller models being capable of semi-autonomous movement, capable of dealing with heavier weather and sea conditions and usually with an autonomy greater than 6 hours. The most common on-board sensors, besides those used in other USVs, are Inertial Navigation Systems and special mission packages[88],[82]. Some examples are the Calzoni U-Ranger (Figure 2-52), the Sea Hunter and the Silver Martin.

Figure 2-52 - Calzoni U-Ranger

Also used in the ICARUS Lisbon Sea Trials, June 2015

2.4.3. **Unmanned Ground Vehicles (UGVs)**

An Unmanned Ground Vehicle or UGV is a mechanized platform which does not carry a human being controlling it, and moves across land[91]. To execute its tasks, it needs to apply techniques of obstacle detection and avoidance that allows road-area detection and recognition, identifying objects of interest, and hazard avoidance during off-road navigation[92],[93]. The terrain where it operates can have various interferences like fences, soft terrain, hills and infrastructures, so it needs to have a strong skeleton to withstand all the bumps and shocks at various speeds required for each mission[94],[95]. It can be classified by its characteristics such as locomotion mode (which can be wheels, tracks, legs and articulated body), type of control system, or weight (Table 2-2).

Light UGVs are typically small, lightweight devices that can be carried and deployed by a small team or only one person, while smaller vehicles can be handheld. They are mostly tele-operated, with some models being capable of limited autonomous movement within small distances of the operator. Their endurance varies from a few minutes to around 2 hours depending a lot on the desired speed and mission and their maximum payload capacity is around 100 kg. The most common on-board sensors are video cameras, IR cameras, collision detection sensors and small manipulator arm control systems[82]. Some examples are the ICARUS light UGV (Figure 2-53), and the Powerbot.

Figure 2-53 – ICARUS light UGV

Photographed during the trials in Marche-en-Famenne 2015

Medium UGVs are typically larger and heavier devices that can carry a wide variety of payloads and require more complex transportation logistics. They are mostly tele-operated and can be deployed at further distances from the operators. Their maximum endurance is of around 6 hours depending a lot on the type of engine and size of the fuel tank (if applicable) and they can carry up to 1 ton of payload. The most common on-board sensors besides those already used in Light UGV, are night vision cameras, LIDAR and chemical and explosive detection sensors[82]. Some examples are the REDCAR, the Ironclad (Figure 2-54) and the RONS MK3 Mod 0.

Figure 2-54 - Ironclad.

Courtesy and copyright of BAE Systems

Heavy UGVs are larger vehicles that vary greatly in size and function. Some may be standard vehicles that use an autonomy kit so substitute the driver/operator and can be reconfigured to allow standard human operation when needed and are thus called "optionally piloted". Their autonomy varies but will typically be around 8 hours (a working day) and can carry payloads of over a ton. The most common on-board sensors are the same as other UGVs, thermal cameras and variable type of mission packages[82]. Some examples are the G-NIUS Guardium (Figure 2-55), the TAGS-DM and the Deployable Universal Combat Earthmover (DEUCE).

Figure 2-55 – Guardium

Source: [96]

2.4.4. **Unmanned Underwater Vehicles (UUVs)**

An UUV is a vehicle that operates under the water surface, like a submarine. It needs a thruster system for propelling the body, and usually fins to allow the platform to ascend and descend[97],[98],[99]. These vehicles can be classified according to the depths they can achieve, their propulsion system (gliders, biomimetic, or classical), or by weight (Table 2-2).

Light UUVs are small and portable vehicles that can be carried and deployed using simple systems and few individuals. Since underwater communication can be complicated, most UUV are quite autonomous, hence the traditional designation of AUV. Their endurance goes from a few minutes to around 12 hours and the average maximum payload capacity is around 30 kg. The most common on-board sensors are SONARs, video cameras, water quality and other water parameter sensors, altimeters, speed sensors and environmental sensors [82]. Some examples are the Gavea, the Robonoise, and the SeaCon (Figure 2-56).

Figure 2-56 – SeaCon.

Developed in partnership between FEUP and the Portuguese Navy

Medium UUVs are typically larger vehicles that sacrifice portability for greater payload capacity, depth, and endurance. Their endurance rounds 24 hours and the maximum payload capacity is about 150 kg, varying a lot with the type of mission. The most common on-board sensors are the ones used on Light UUVs, acoustic modems, Satellite Communications (SATCOM) at the surface, acoustic imaging, more advanced SONARs and other environmental sensors[82]. Some examples are the ATLAS SeaCat, and the REMUS 600 (Figure 2-57).

Figure 2-57 - REMUS 600

©2018 Hydroid, Inc.

Heavy UUVs are typically described by being larger, more capable vehicles that may use modules or be designed to conduct specific tasks. Their endurance is usually more than 20 hours and they can carry more than 150 kg of payload. The most common on-board sensors besides those used in other classes of UUVs, are advanced SONAR arrays and special mission packages[82]. Some examples are the Boeing Echo Ranger, the i-Tech 7 QX Ultra, the Proteus, the ISE Ltd. Theseus (Figure 2-58), and the USN Large Vehicle Class UUV (planned).

Figure 2-58 – Theseus

Courtesy International Submarine Engineering Ltd

2.4.5. **Unmanned Hybrid Vehicles (UHVs)**

There are some vehicles that are designed to operate in multiple environments (air, ground, surface, and underwater) and are called Unmanned Hybrid Vehicles (UHVs). The design of these vehicles tends to be challenging, since they're aimed to function and adapt to completely different environments. For instance, employing an adequate propulsion for an aerial and underwater environment would require the use of propellers, instead of other means of propulsion due to its better performance for both aerial and underwater tasks[100]. However, propellers for air and water have completely different characteristics. Allowing for an unmanned vehicle to operate on both air and ground (for example) provides versatility, proving that the vehicle can do aerial reconnaissance over large areas, followed by payload delivery and operation over ground allowing it to enter structures and to examine them at close range[101]. The primary difficulties of these vehicles are the transition between mediums as well as the landing, taking off and thrusting[102].

2.5. **Missions**

UxVs were traditionally given the DDD missions: Dangerous, Dull, and Durty. They have evolved and are now used in a very wide variety of activities.

UxV have the advantage of being able to execute hazardous missions without putting the operator in harm's way. This is particularly important for missions that involve handling radioactive and explosive components, or missions where someone may try to interfere.

The fact that they can be quite small allows them to execute missions in hard to reach areas that would be inaccessible to human operators or manned systems. They can also be used when it is necessary to avoid detection, since they can have very silent propulsion systems.

Also, UxV can perform repetitive and dull missions without decaying effectiveness over time. In terms of mission cost, UxS are seen as cheaper than their manned equivalents [82], although there is some debate regarding this.

UxS missions/tasks can be divided in Military missions and Civilian missions.

2.5.1. **Military Missions**

The main military missions of UxS, defined *e.g.* in the Portuguese Navy's Concept of Operation for UxV where the author was involved, and that drew inspiration form NATO's concept of operation, presented in (Figure 2-59) are the following:

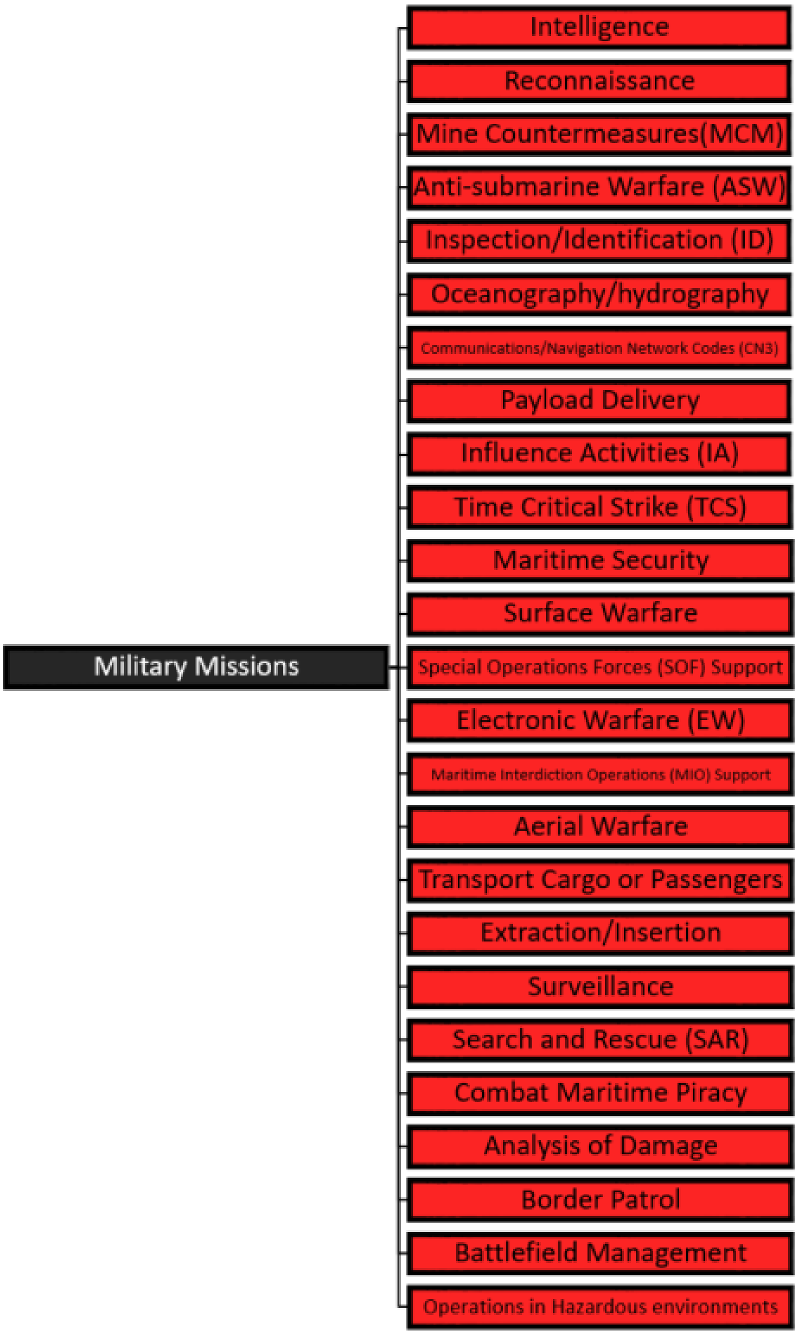

Figure 2-59 - Military Missions

- Intelligence: gathering, analysis, protection and dissemination of information about the enemy, terrain and weather in an area of operations or area of interest[103];
- Reconnaissance: inspection or exploration of an area to gather information[103],[104];

- Mine Countermeasures (MCM): operations in minefields as an off-board sensor while the host ship stays outside the minefield boundaries[105];
- Anti-Submarine Warfare (ASW): robust tracking of quiet diesel electric submarines[103];
- Inspection/Identification (ID): support for Homeland Defense (HLD), Anti-Terrorism / Force Protection (AT/FP), and Explosive Ordnance Disposal (EOD) needs[106];
- Oceanography/Hydrography: collection of environmental data that directly supports anti-submarine, mine, amphibious, strike, special and expeditionary warfare[107];
- Communication/Navigation Network Codes (CN3): connectivity across multiple platforms, as well as navigation assistance on demand[103],[108];
- Payload Delivery: clandestine method of delivering logistics to support a variety of other mission areas. The missions supported include MCM, CN3, ASW, Oceanography, Special Operations Forces Support, and Time Critical Strike (TCS)[106];
- Influence Activities (IA): deception, deterrence and disruption of enemies[104];
- Time Critical Strike (TCS): delivery of ordnance to a target with sensor-to-shooter delay measured in seconds, rather than minutes or hours[109];
- Maritime Security: security of allied domestic ports, waterways, and protection of ship and maritime infrastructures (piers, docks, anchorages, warehouses) against a spectrum of threats from conventional attacks to special warfare or specifically targeted terrorist attacks[104];
- Surface Warfare: armed engagement of threats in open waters, as well as littoral warfare[103];

- Special Operations Forces (SOF) Support: missions involving unconventional warfare, counter-terrorism, reconnaissance, direct action and military assistance[108];
- Electronic Warfare (EW): means of deception, jamming, and warning of electronic attacks[106];
- Maritime Interdiction Operations (MIO) Support: diversion, disruption, delaying, or destruction of the enemy ‘s merchant marine trade. Drug interdiction and alien migrant interdiction operations[104],[109];
- Aerial Warfare: engagement of aerial threats[104];
- Transport Cargo or Passengers: delivery of cargo or passengers in dangerous environmental conditions[105];
- Extraction/Insertion: payload extraction/insertion from/to a specific target or location[106];
- Surveillance: monitoring of the behaviour of people, objects or processes for conformity with expected or desired norms [104];
- Search and Rescue (SAR): search for, and provision of aid to people who are in distress or imminent danger[106],[105];
- Combat Maritime Piracy: combat violence or plunder on the high seas or in the air, for private ends, using aircraft or vessels[104];
- Analysis of Damage: collection of data (images and video) related to disasters, or the effects of attacks[104];
- Border Patrol: monitoring, regulation or control the movement of people, animals and goods into or out of a country [106],[108];
- Battlefield Management: improving of the Command and Control capabilities of a force commander in the field[110];

- Operations in Hazardous environments: operations where humans couldn't operate (CBRN environments, high pressure environments, crumbling buildings)[111],[112].

2.5.2. Civilian Missions

UxVs may be used in a wide variety of areas and for various missions, that can be classified as follows (Figure 2-60):

Figure 2-60 - Civilian Missions

- Monitoring: monitor the behaviour of crowds, traffic, animals, tree growth, pollution and air sampling[113];

- Scientific Exploration: validation of geological surveys, helping researchers gain a deeper understanding of the environment[114];
- Agriculture and Forestry: monitoring of crops and pesticide spraying services[113];
- Aerial Photography: taking photographs and filming various events[113];
- Engineering and Construction: performing inspections on power transmission lines and high-voltage towers[113];
- Navigation: identification of reference points that aid navigation[113];
- Law Enforcement: incident surveillance, security, drug enforcement and search for missing people[103];
- Fire Service and Hazardous Materials Operations: fire detection, incident control and dangerous material handling[115], [109];
- Emergency Medical Services: provision of medical assistance through the transportation of first aid kits and other medical material[115];
- Support River Authorities: river monitoring, flood and pollution control[116];
- Meteorological Services: weather forecast through the analysis of samples[113];
- Wildlife and Fisheries Management: animal and fisheries protection. Marine environmental protection[113];
- Site Security: monitor pipelines or other installations to keep them secure from tampering[103];
- Surveying: Geographical, geological and archaeological survey[103];

- Disaster Response and Damage Assessment: disaster control, cooperation in search and rescue operations and visual assessment of damaged areas[116];
- Marine Environmental Protection: oil spill response, identification or removal of marine debris, among others[117];
- Disaster Management: Real-time communication assistance, assessment and mapping of damage, pre- and post-event monitoring[116];
- Environmental Survey and Measurement: cloud and aerosol measurements, measuring of carbon dioxide flux, water vapor and total water measurements, coastal ocean observations, O2 and CO2 flux measurements, estimation of glacier and ice sheet dynamics[118], vertical profiling, heating rates, ice sheet thickness and surface deformation, measuring of cloud properties, physical oceanography, meteorology and support of atmospheric chemistry;
- Focused Observations in Extreme Weather: observations and recording of information of extreme weather events like hurricanes[119].

3

Interoperability Building Blocks (IBB)

This chapter introduces the more relevant existing communications methods, standards, data models, frameworks, and protocols used in the area of autonomous systems and make some comparisons between them. We chose to call all of these “Interoperability Building Blocks”. The reason for introducing this new concept is to have a common term to refer to all these entities, already introduced and explained in chapter 2, putting an emphasis on the fact that they all contribute to achieve the desired interoperability.

3.1. Most Relevant Interoperability Building Blocks

There are many communication methods, standards, data models, frameworks, protocols, reference architectures, etc., that can contribute to increase interoperability of UxS at different levels. None of them is a “silver bullet” that solves all problems, and some address very specific issues, but they must all be taken into account if we want to have an overarching view of interoperability. The IBBs chosen are all widely used in the UxS community, and in the military community in particular.

3.1.1. Standardization Agreement (STANAG) 4586: Standard Interfaces of UAV control systems (UCS) for NATO UAV interoperability

NATO STANAG 4586 standardizes interfaces of UAV control systems within NATO. Its development was done by a group of specialists from various NATO countries, and started in 1998. The original version was approved in 2004, and the second in 2005. The third version[120], in 2012 introduced changes that were not always backward compatible, and many suppliers did not adhere to it.

Discussion are currently ongoing to approve version 4 (the author of this thesis has been involved in those meetings), which will hopefully reunite the community providing full backwards compatibility with versions 2 and 3.

This STANAG allows a much smoother and easier information sharing, which is very important in war fighting capability of the forces. The purpose is to have interoperability between UAVs, GCS and the Command, Control, Communication, Computer and Intelligence (C4I) segments of the system to work in a NATO environment. The aim of this STANAG is not to achieve an operational improvement of the UAV systems, but only to increase interoperability in the communication with the UAV Control System (UCS) within the allied forces[121],[122].

The UAV system is divided into five elements (Figure 3-1) which are: the air vehicle, the payload, data link, UCS and finally launch and recovery system[123].

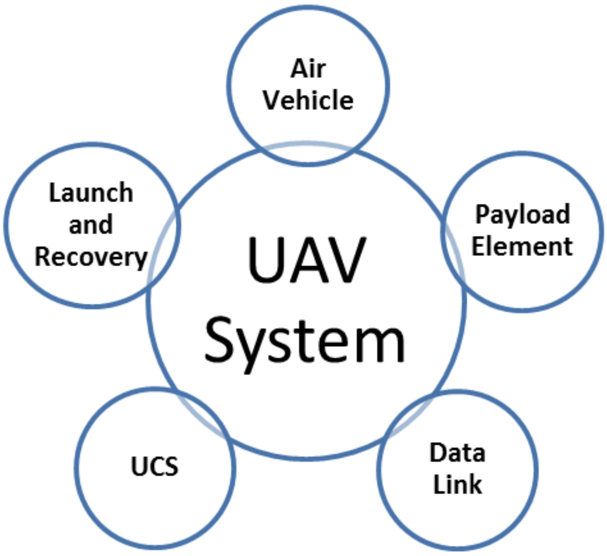

Figure 3-1 - Elements of the UAV System in STANAG 4586

The *Air Vehicle* element includes the propulsion, avionics and every other element that helps flight management aboard the vehicle. The *Payload* element consists of all the units that are associated to the mission, for example weapon systems or specific cameras. The *Data Link* is responsible for the communication and is divided into the *Vehicle Data Terminal* (VDT), in the air vehicle, and the

Control Data Terminal (CDT) in the GCS. The *UCS* is responsible for mission control, including the C4I segments. The *Launch and Recovery* element is responsible for the launch and recovery of the vehicle.

Defining the interfaces that should be implemented to achieve the required Level of Interoperability (LOI) according to the assumed Concept of Operations (CONOPS) is the primordial objective of STANAG 4586. It will only be possible through the implementation of standard interfaces in the UCS to communicate with different UAVs and their payloads[124]. The implementation of standard interfaces will also simplify the integration of components from different origins (vendors) as well as the interoperability with legacy systems. Compliant UAV´s shall be certified and will increase NATO joint flexibility through the sharing of assets[125].

There are five levels of interoperability in this standard, to accommodate different operational requirements. The respective operational requirements and CONOPS will determine or drive the required LOI that the specific UAV System will achieve.

- Level 1: Indirect receipt and/or transmission of sensor product and associated metadata.
- Level 2: Direct receipt of sensor product data and associated metadata from the UAV.
- Level 3: Control and monitoring of the UAV payload (and only payload) unless specified as monitor only.
- Level 4: Control and monitoring of the UAV, unless specified as monitor only, less launch and recovery.
- Level 5: Control and monitoring of UAV launch and recovery unless specified as monitor only.

This standard establishes the following elements and interfaces: Air Vehicle (AV), Vehicle Specific Module (VSM), Data Link Interface (DLI), Core UCS (CUCS), Command and Control Interface (CCI), Human Computer Interface (HCI), Command and Control Interface Specific Module (CCISM)[126]. These elements are illustrated in Figure 3-2.

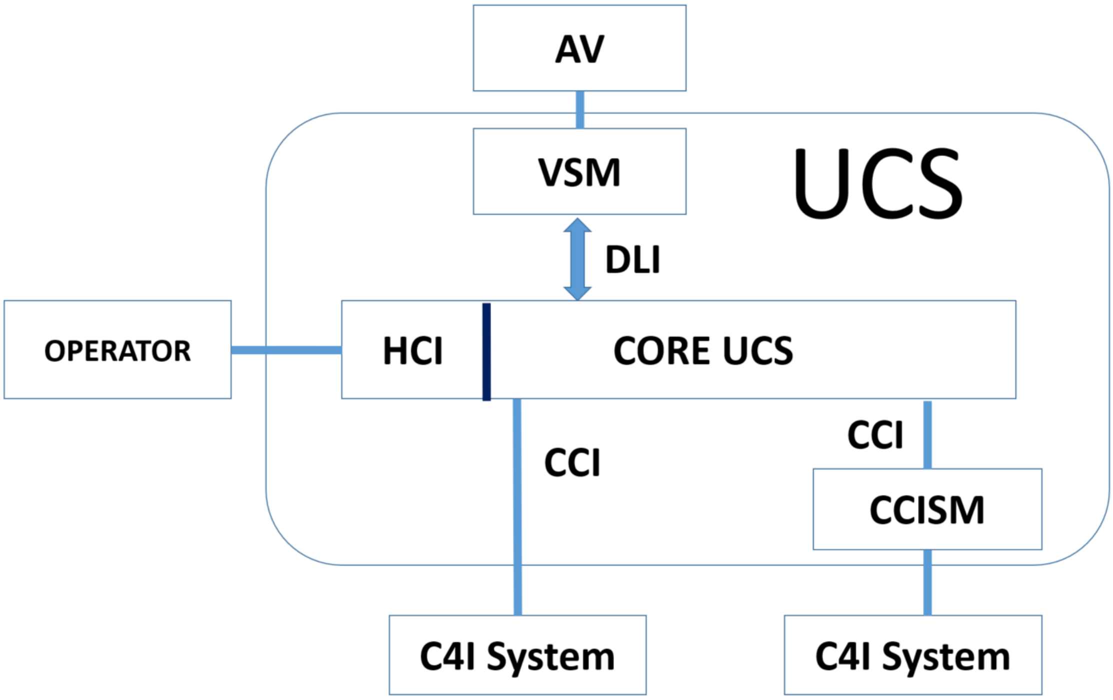

Figure 3-2 - STANAG architecture

The VSM, according to the AV requirements, provides unique/proprietary communication protocols, interface timing, data formats and rendition of the DLI protocols and message formats[127].

The DLI enables the CUCS to generate and read specific messages for control and status of air vehicles and payload. DLI specifies the mechanism to process and display specific messages, independent of the AV and payload.

The CUCS provide a user interface that enables the operator to conduct all phases of an UAV mission, and support all settings from the DLI, CCI and HCI. The computer generated graphic user interface should also enable the operator to control different types of UAVs and payloads[128].

CCI establishes the standard message set and concomitant protocols that have been selected to be C4I system/node independent to cover all types of messages and data that need to be exchanged in all the phases of a UAV mission [127].

The HCI establishes the operator display and input requirements that the CUCS shall support. Although not specifically defining the format of the data to be displayed, there are some identified requirements that the CUCS shall provide to ensure an effective operation of the UAV system.

The CCISM, like VSM, encapsulates the CCI data and any translation required to be compatible/interoperable with the physical communication links between the UCS and the C4I systems[126].

STANAG 4586 is probably the most widely used standard in large UAS, and it is used in conjunction with a number of associated STANAGS so as to provide a usable system. These associated STANAGS include 4545 for Secondary Image Formats, 4575 for Data Storage Interface, 4607 for Ground Moving Target Indicator Format, 4609 for Digital Motion Imagery Standards, 7023 for Air Reconnaissance Primary Imagery Data Standards, 7024 for Imagery Air Reconnaissance Tape Recorder Standards, 7085 for Data Links for ISR Systems, 4559 for the Standard ISR Library Interface, and 4670 for Training of Designated Unmanned Aerial Vehicle Operators.

Given the importance of this STANAG, we used it quite a lot and, in later chapters, we will describe how we developed STANAG to JAUS interfaces. An example of a STANAG 4586 message is given in Figure 3-43, and as can be seen it is quite long, even for a simple command.

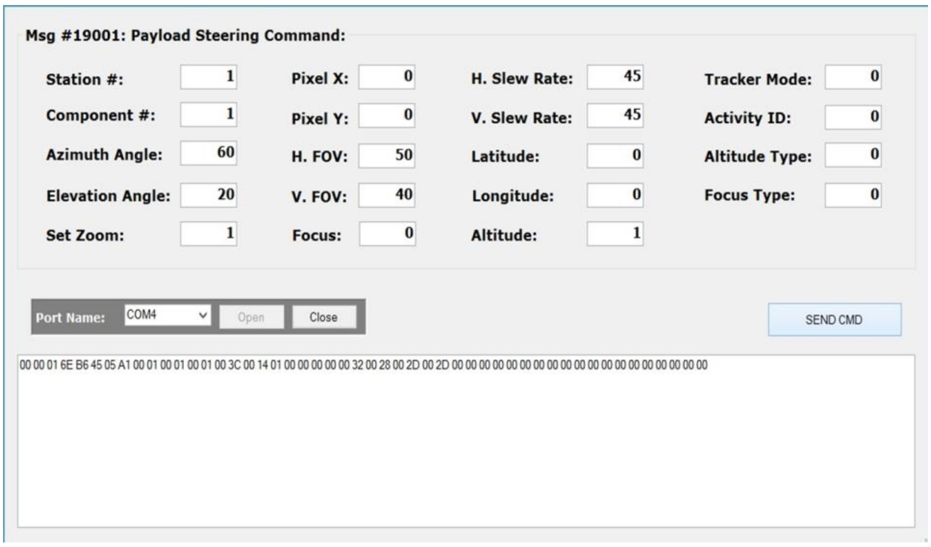

Figure 3-3 - An example of a STANAG 4586 message.

Example of a computer program, used by the author in his lectures, that generates STANAG commands (in the large bottom white window), given a number of parameters introduced by the user. In this figure we can see the string of characters that orders a camera to look in a given direction.

3.1.2. Joint Architecture for Unmanned Systems (JAUS)

In 1998, the Office of the Under-Secretary of Defense (OUSD) for Acquisition, Technology and Logistics Joint Robotics Program commissioned a working group to design a standard for interoperability of UGVs. The standard was then called Joint Architecture for Unmanned Ground Systems (JAUGS), but later was generalized for unmanned systems of all sorts. In 2005, JAUS was adopted by the Society of Automotive Engineers (SAE) as a standard under the auspices of its aerospace standards division[125],[5]. The objective of JAUS is to make the communication between robots more efficient, by reducing communication times, with a standard that promotes interoperability. Formally SAE is still responsible for the standard[129] and the documents that define it have to be purchased from them. However, a number of "branches" have emerged, mainly due to successful and widely available implementations of the standard. Of these, the most relevant is probably OpenJaus[130].

So as to assure interoperability over a wide range of UxS, there are number of important characteristics that JAUS has be design. In particular, it is:

- Platform independent, so that it can be used on any type of vehicle;
- Mission independent, so as to be successfully used and capable in a vast range of tasks or environments, being as robust as possible;
- Computer hardware independent, as there are several types of system and sensors that can be used. The growth rate of the computer industry is high, and the standard cannot require a specific hardware implementation, as it would require constant updates to the standard. Also, each UxS has its own hardware, depending on the developer and on the mission requirements. Therefore, it is important for JAUS to be hardware independent;
- Technology independent, at a higher level than the simple hardware level referred before. To withstand technological evolution, JAUS cannot specify a specific technology, as there will inevitably be several solutions to each single problem;
- Allow operator empowerment, as JAUS intends to let the operator decide the best approach for a certain problem.

The standard has different hierarchical levels as can be seen in Figure 3-4. Each one of these has its own nomenclature. At the first level, there is a *System*, which provides the union of all robotic capabilities. Systems are composed by various *Subsystems*. These subsystems are responsible for one or more functions and each one of them has its own communication, command and control tools. Subsystems can be autonomous vehicles. Each subsystem is formed by *Nodes* (computer processors) which are responsible for a set of functions. Nodes are formed by *Components*. Components provide one unique capacity for the system and they can be, for example, an application or a thread running a service[131].

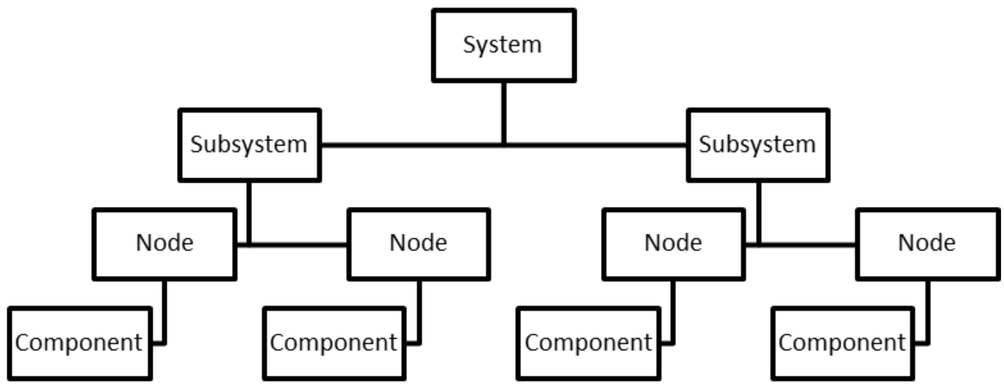

Figure 3-4 - JAUS architecture

Basic configuration of JAUS architecture, with 4 hierarchical levels: System, Subsystem, Node, and Component.

A Service simply provides some useful function for the system. The Service Oriented Architecture (SOA) enables distributed command and control of the UxS. The SOA approach of JAUS formalizes the message format and protocol interaction between system components[132]. This approach is standardized by the JAUS Service Interface Definition Language (JSIDL), an XML-based language that provides the basic standard and syntax for specifying JAUS Services. All the Services that are standardized by JAUS must be specified in valid JSIDL syntax [133].

There are already many documents published by SAE that specify JAUS procedures. They can be divided (Figure 3-5) in: JAUS Transport Considerations specifies; JAUS/SDP Transport Specification; JAUS Messaging over OMG Data Distribution Service; JAUS HMI Service Set; JAUS Compliance and Interoperability Policy; JAUS History and Domain Model; JAUS Core Service Set; JAUS Mobility Service Set; JAUS Manipulator Service Set; JAUS Service Interface Definition Language; JAUS Unmanned Ground Vehicle Service Set; JAUS Mission Spooling Service Set; JAUS Environment Sensing Service Set.

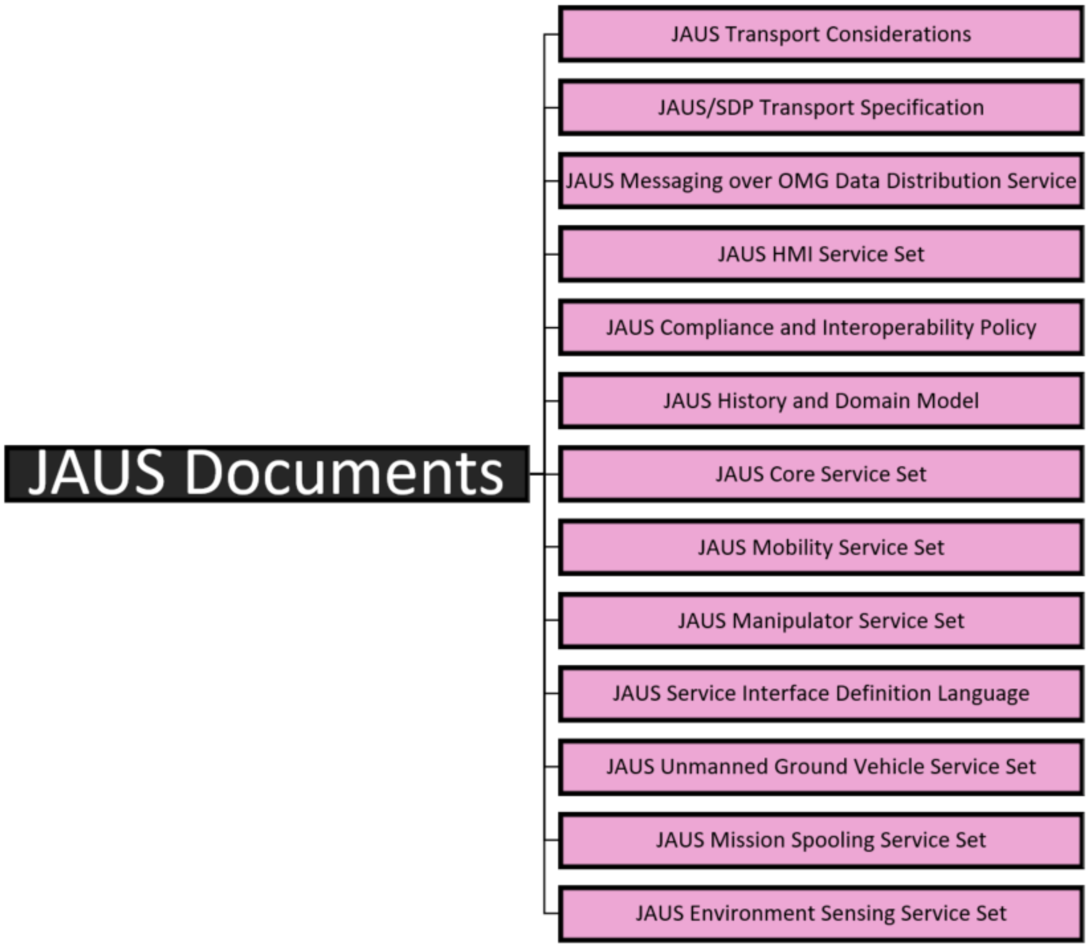

Figure 3-5 - The JAUS Documents that constitute the standard

JAUS Transport Considerations[134] specifies how JAUS's messages are transported, including the media infrastructure used and how it interfaces with the rest of the system. This is further detailed in another document *JAUS/SDP Transport Specification*[135], that explicitly defines the protocols that may be used. A third document related to this, the *JAUS Messaging over OMG Data Distribution Service(DDS)*[136] defines the standard representation of the messages in DDS IDL defined by the Object Management Group (OMG) CORBA 3.2 specification.

JAUS HMI Service Set[137] specifies the Human Machine Interface (HMI) capabilities, since JAUS explicitly considers that the user may give commands using a pointing device, a keyboard, drawings, and generic digital and analog control devices. *JAUS Compliance and Interoperability Policy*[138] recommends an approach to documenting the complete interface. *JAUS History and Domain Model*[139] gives implementers the historical background and justification for the choices taken, and in part describes the underlying reference model and operational concepts that were used to develop JAUS. *JAUS Core Service Set*[140] defines common services to be used in all components. *JAUS Mobility Service Set*[141] standardizes command and control services that are vehicle independent. *JAUS Manipulator Service Set*[142] defines a message-passing interface to enable interoperability between communicating elements in the unmanned system. *JAUS Service Interface Definition Language*[143] guarantees the validation and efficiency of messages and service data structures. Each JAUS service has its own defining document, known as JAUS Service Definition (JSD), that is expressed in this language. *JAUS Unmanned Ground Vehicle Service Set*[144] defines the platform-specific capabilities for UGVs. *JAUS Mission Spooling Service Set*[145] defines the message-passing interface for mission spooling services. The *JAUS Environment Sensing Service Set*[146] defines procedures associated with the message-passing interface for commonly used sensors.

The JAUS standard is built upon JSIDL which defines an XML schema that enables formal specification of JAUS Services and Messages. This schema assists interoperability by removing some of the ambiguities that can plague other standards.

To define the interoperability between different systems, JAUS defines three levels of compliance:

- Level 1 requires that all messages between subsystems be in JAUS. If for example, a certain vehicle architecture is not JAUS, it has to have a JAUS adapter;
- Level 2 is achieved when all messages between subsystems and nodes are JAUS;

- Level 3 requires that every message between subsystems, nodes and components must be in JAUS.

JAUS has many advantages when compared to other standard. As previously mentioned, it is mission, computer hardware, technology and platform independent. It can be used in very different systems, and still assure interoperability. Different subsystems can be developed in different programing languages or development system, and they can still integrate the standard at any moment. Also, as it is modular, one component failure does not cause the failure of the system[147].

However, JAUS also has some disadvantages. Because it defines messages for each component, it has a pre-defined set of components for each subsystem. This may limit the options of the developer, although a new component may be added (but not recognized by other entities). Also, it is necessary to have knowledge on the assignment of subsystems and nodes IDs to address messages, since there isn't any self-discovery mechanism.

In conclusion JAUS is a standard that has very good acceptance in the UxS community, as the advantage and disadvantage relation is very positive in comparison with other standards.

The reference architecture proposed in this thesis is strongly influenced by this standard, since it was used with considerable success in the most relevant research projects where we were involved.

3.1.3. **Mission Oriented Operating Suit (MOOS)**

MOOS was created by Paul Newman between 2001-2005, with the help of students and researchers at Oxford and at the Massachusetts Institute of Technology[148],[149],[150]. MOOS is particularly popular in the maritime robotics community. It is an open-source implementation and was originally designed to provide an autonomous helm. It uses a publish/subscribe philosophy, and substantial support applications are available, that make it quite popular in USVs and UUVs.

The creation of MOOS was done taking into account some main goals: it should be platform independent; control processes should run the vehicle (each one specializing in a single function); and communications should be robust and fault tolerant.

There are lots of MOOS Applications (MOOSApp) in the MOOS community and each one of them is connected to a single MOOS Database (MOOSDB) that is in the center of the whole system, as is shown in Figure 3-6.

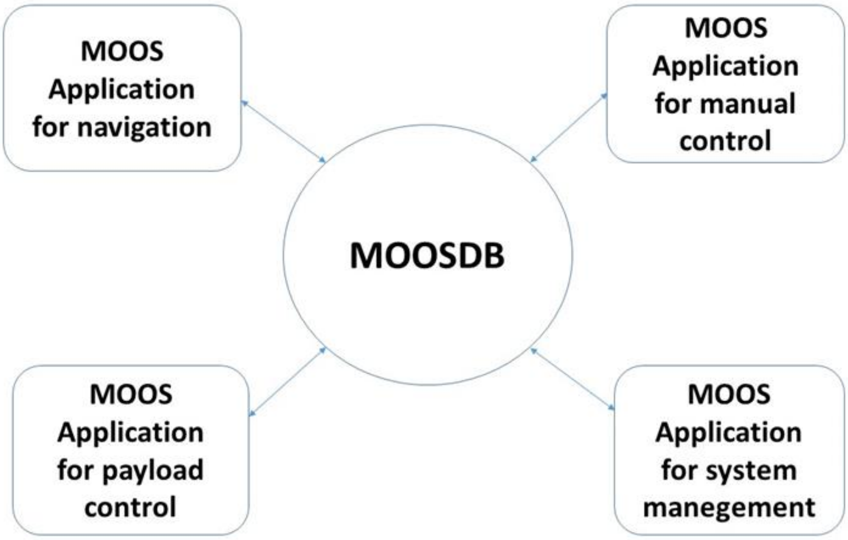

Figure 3-6 - MOOS Functional Standard

Therefore, MOOS have a star-like topology. The network has some properties including: no peer to peer communication; each communication between the client and the server is started by the client; each client has its own name; one client does not have to know about the existence of another client and therefore can't communicate with other clients; the network can be distributed over other systems, if they run the supported operating systems[151].

The key idea with respect to facilitating code reuse is that applications are largely independent, defined only by their interface, and any application is easily replaceable with an improved version with a matching interface. The MOOS Core includes the MOOSDB application and the MOOS application superclass that each individual MOOS application inherits to allow connectivity to a running MOOSDB. Since the MOOS Core and many common applications are publicly available along with source code under an open-source General Public License (GPL), a user may develop an improved module by altering existing source code and introduce a new version under a different name. Holding the MOOS

Core part of the code-base constant, between MOOS developers, enables the plug-and-play nature of applications[152].

In order to build a MOOS community (*i.e.* a set of systems that use MOOS), it is necessary to use MOOSDB and two libraries: MOOSLib and MOOSGenLib. These libraries contain all the functions that the client needs to build his system. MOOSLib's primary objective is to provide communication components and configurations that are a baseline for most of the applications. On the other hand, MOOSGenLib contains *utilities* and classes that are used throughout MOOS. This library can provide platform-independent serial ports, threads for safe configuration reading tools, string manipulation or parsing tools, among other useful tools[153],[154],[155].

To conclude, there are some advantages and disadvantages in MOOS. One of the disadvantages is that the centralized topology makes it vulnerable to "bottle-necking" (vulnerable to congestion). Although this is true, there are lots of advantages with this system. No matter how many participating clients are in the network, it remains simple. The server knows all active connections and it is responsible for the allocation of communication resources. The client is independent from other connections between the server and other clients, which prevents the interference with others. Finally, it also has wide support within the research community, notably from NATO's Centre for Maritime Research and Experimentation (CMRE).

3.1.4. **Compact Control Language (CompactCL)**

CompactCL was released in 2005 by the Woods Hole Oceanographic Institution (WHOI) and it is a standard created in to allow the communication between multiple UUVs and a central point in an efficient manner. It also allows these vehicles to communicate with each other. The objective of this standard is to allow systems to communicate through acoustic links which have a limited bandwidth, and thus needs to be extremely compact. It was initially developed for REMUS 100 and its derivatives, but it can be applied to other vehicles. REMUS 100 is one of the most important UUVs in the market and it is used in marine research, defense, hydrographic and offshore/energetic markets.

This standard is designed based on the capabilities of WHOI Utility Acoustic Modem (UAM) and the WHOI Micro-Modem acoustic communication system[156],[157] and so it was designed to have 32-byte packets. Therefore, this standard includes messages for control, sensor and status information, which are compressed into 32 bytes, but does not include image information, although it can be added in future developments.

CompactCL messages cover various categories[158]:

- Vehicle information such as position, heading, speed and subsystem fault status;
- Standard data such as bathymetry;
- Special messages such as those generated when a computer-aided detection system finds an object of interest in a side-scan SONAR record;
- File transfer with acknowledgement. The format of the messages is such that it can be interleaved with other types of messages and with multiple vehicles. Each communication transaction includes a short network packet that specifies the source, destination and data rate of the packet to follow.

This standard does not cover error detection or correction in a message, because it is assumed that this problem is dealt with in the transport layer, which is true in the WHOI modems that were referred previously. Thus, in this standard it is assumed that all messages transmitted are received. This can be a problem with critical information such as redirection commands, but the standard supports an acknowledge bit. There are also no fields indicating source or destination addresses. Therefore, this work is done by modems. Finally, there is no priority sending data, so this is completely under the control of the transmitter.

Using simple network modes, those messages can be broadcast from a command center, or sent from vehicle to vehicle, as in the example of Figure 3-7[159].

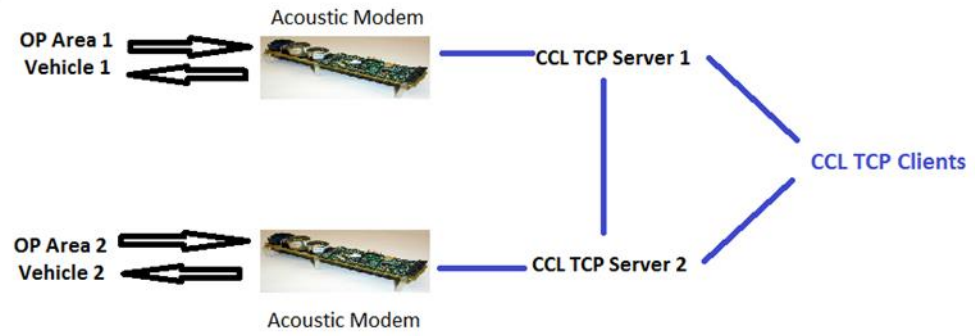

Figure 3-7 - Connection between nodes of a CompactCL System

The CompactCL libraries are written in C, and the messages are defined in C "struct" constructs, but they can easily be defined in Pascal, Java or other programing languages that have similar constructs.

Since CompactCL is basically a simple point-to-point protocol to transmit short messages, an improved standard in matters of flexibility and reconfiguration was developed, named the Dynamic Compact Control Language (DCCL). DCCL, although based on CompactCL, can be used in a network of devices and is easier to reconfigure because it is designed in XML[160].

In conclusion, this standard is very important for acoustic systems, and it presents some advantages and disadvantages. The low bandwidth is one of the main advantages, as it allows efficient communication between UUVs that have to use underwater acoustic links. Another it that is natively supported by the popular micro-modem acoustic systems. Finally, it can easily be used (and frequently is) just for the acoustic links of a more encompassing standard. As for the disadvantages of this standard, the fact that it is aimed at UUVs, with low bandwidth, makes it a bad option for joint forces with other types of UxVs, such as UAVs or UGVs, that will normally require high bandwidth. Also, in practice, it requires the use of WHOI micro modems, since other vendors do not support it. Nevertheless, in our Navy we use in some of our UUVs.

3.1.5. **Common Control Language (CommonCL)**

CommonCL was developed in 2003 by the Office of Naval Research, specifically for autonomous underwater vehicles communication and control. With the technology development, more and more UUVs are being created and because of this there are lots of heterogeneous standards, making the communication between different systems very difficult. Its design aims to provide a standard vocabulary and grammar for inter-UUV and UUV-human communication[161], [162],[163].

The objectives of CommonCL were to create a standard that would answer 3 fundamental questions: firstly, that would be a descriptive standard to be used by a controller of a vehicle; secondly a standard that could improve with the development of new applications; and finally, to provide an interpretation of mission specification. So, it would be a standard used for communication between UxS and for coordinated tasks.

CommonCL has the following design requirements[164]:

- One vehicle isn't allowed to *look* inside another vehicle;
- Cooperation occurs only through message-passing between the decision-making levels among platforms;
- Users should be able to add their own messages if required but can't expect that these new messages will be understood by all vehicles, because they are not part of the standard;
- Allow for different execution behaviors, *e.g.* repetition, sequential or parallel, as well as support "canned missions" and interactive tasking; facilitate the extensibility to new vehicles and new missions;
- Build upon previous work on generic behaviors as well as other UUV and intelligent agent development efforts;
- Optimized for the UUV domain;
- Optimized to conserve transmission bandwidth.

CommonCL has a detailed list and description of these commands. One of the characteristics of CommonCL is the small size of serialized form of the messages due to be targeted to be used in acoustic communications[165].

The standard of CommonCL is divided in five layers (Figure 3-8): vocabulary and message set specification; CommonCL support library; basic behavior process set; mission interpreter; cost-based real-time planning[166].

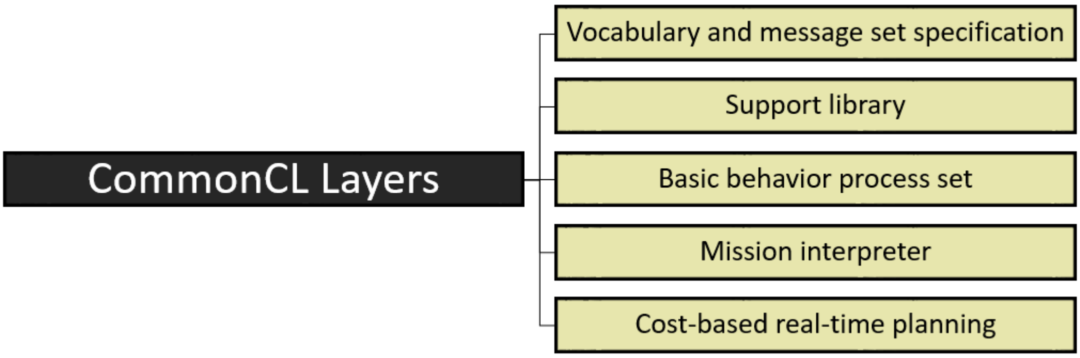

Figure 3-8 - CommonCL Layers

Vocabulary and message set specification is, like the name indicates, the format of messages and it defines the specific domain vocabulary.

CommonCL *support library* defines the implementation of the vocabulary and messages, optimizing the messages for low bandwidth.

The layer of *basic behavior process set* specifies a standard for managing behavior processes and provides software processes to interpret CommonCL messages, which interface to both vehicle-specific and high-level problem-solving processes.

The *mission interpreter* layer provides the C language-like grammar for mission level development, the automatic generation of executable behaviors based on a mission file and the ability to update tasks in real-time.

The *cost-based real-time planning* is the adaptive re-planning which allows optimization of tasks to cope with dynamic aspects of the environment, working toward individual and potential group goals[167].

As previously stated, this standard can optimize steps that are ahead and update tasks that are received in real-time. Because vehicles are different, they may have different capabilities for different tasks, CommonCL determines who has the better capabilities for each specific mission. This turns the solution much more efficient, allowing the coordination between vehicles.

This standard focuses in some basic behaviors, as previously stated:

- *Maneuver* which are the primary functions that will allow the vehicle to move to a new position;

- *Maintain position*, where the UUV keeps in the same position, usually doing circles and trying to save as much energy as possible;
- *Navigate* where the system specifies path constraints and monitors the actual position while the UUV maneuvers.

In CommonCL the control station can issue five types of request to a vehicle, to obtain:

- Status Information, which can be done using a single request or requesting a periodic update of basic information about the vehicle (speed, depth, battery level, etc);
- Capabilities, *i.e.* a list of the UUV's main systems, sensors and actuators;
- Files, where it is requested that the UUV send a specific file;
- Parameters, where it is requested that the UUV send a value of a specific parameter;
- Configuration changes, which allows the UUV to change some preconfigured values.

To conclude, CommonCL has the following advantages when compared with other standards: it is targeted and optimized for UUVs; it provides an efficient way of exchanging information between UUVs and between these and ground stations. However, CommonCL also has some major disadvantages. Because it was created for underwater operations, this standard cannot be used efficiently for data exchange with different types of vehicles, such as UAVs or UGVs, that normally have a much higher bandwidth requirement.

3.1.6. **Coupled Layered Architecture for Robotic Autonomy (CLARAty)**

CLARAty was created by National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory in 2000, mainly because of the Mars Rovers. It is a collaborative effort among several institutions: California Institute of Technology's Jet Propulsion Laboratory, Ames Research Center, Carnegie Mellon University, and several other universities and members from the robotics community. CLARAty tries to provide a standard for algorithms developed for robotic systems that can be generalized, while maintaining the ability to easily integrate platform specific algorithms[168].

The CLARAty standard was designed with four main objectives[169],[170]:

- To reduce the need to develop custom robotic infrastructure for every research effort;
- To simplify the integration of new technologies onto existing systems;
- To tightly couple declarative and procedural-based algorithms;
- To operate many heterogeneous UxV with different physical capabilities and hardware architectures.

One of main differences from the other approaches reviewed previously, is the focus on planning, that takes into account a high-level description of goals, using a mainly declarative approach, and breaks it down into tasks that are performed by software objects that in turn interact with the hardware.

The CLARAty standard has two distinct layers: *Functional Layer* and *Decision Layer* (Figure 3-9).

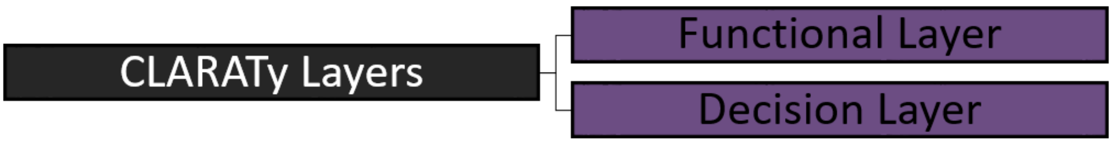

Figure 3-9 - CLARATy Layers

The Functional Layer lies between hardware and the Decision Layer. It is an object-oriented hierarchy, which helps with system abstraction. These objects encapsulate the characteristics of each component of the UxS. All objects can have their own planners for specific tasks and resource usage predictors, which helps with the efficiency of the system. The various objects in the functional layer are described in UML, and each should have simulation capacities and be tested and debugged separately[168].

The Decision Layer receives the goals of the mission and breaks it down into a goal net, using a mainly declarative approach. This net, in turn, is broken down into a task tree, and for this the decision layer must take into account the limitations that the functional layer and the operator identify.

When the system is running, it receives and creates goals while receiving resource prediction's information from the functional layer to perform its task plan. After this scheduling is done, tasks are created, and the task tree is modified if necessary. During execution, feedback from the functional layer may lead to re-planning [171].

Two models of data flow are used in CLARAty: push and pull models. For systems that have bandwidth limitations on a shared bus, and where the need for data is asynchronous and constitutes a subset of all possible information that can be obtained, a pull model allows maximum flexibility. If the usage is predictable and synchronous, then a push model is used. For a given bus, and if both modes are supported by hardware, it is possible to switch between these two modes depending on the system configuration[172],[173].

CLARAty presents some advantages and disadvantages. It is a standard for generic and reusable robotic components, thus allowing any new components to be used. However, the main disadvantage of this standard is the fact that it is specific for UGVs and does not take into account the large variety of UxS used in joint operations[174].

Despite its attractive high-level approach, it has not had widespread support outside the community where it was developed.

3.1.7. **European Component Oriented Architecture (ECOA)**

The ECOA started in 2008 as a collaboration between industrial partners (mainly BAE systems) from the United Kingdom (UK) and from the French Ministry of Defense. It is aimed at aircraft systems (not necessary unmanned) and was motivated by the increasing complexity and costs of military aircraft software systems.

ECOA has measures to reduce development and life-cycle costs of military platforms that have complex software systems. Its improved software architectural approaches allow cooperation and interaction between vendors, so as to achieve maximum operational effectiveness with minimum cost, to support a fleet of platforms[175].

ECOA, uses as main building blocks software components named Application Software Components (ASC). These components may provide services, but they can also require them from another ASC. Each ASC has its own component properties, depending on the type of component, and insertion policies, which define what services it needs to use (from other ASC) to do a certain service[152].

This standard produces a database of ASC, each with a list of services provided, properties, and services (or characteristics) it requires. Developers can create their own services and component interactions, building their own scheme, but can also use pre-existing models and change how they link with each other. A key issue is that the interaction amongst ASC, and between these and the infrastructure (both hardware and low-level software) be kept within the strict boundaries of services, properties, and insertion policies, so that an ASC can be exchanged with one from another vendor (or duplicated to provide redundancy) with minimum implications (probably only in Quality of Service (QoS), that must nevertheless be measurable). In an ECOA system, all the interactions between modules that implement ASC rely on three mechanisms: event, versioned data and request-response. In addition, calls and handlers exist for infrastructure services to allow the management of the runtime lifecycle, logging, faults and time.

The ASC interface is referred as the module interfaces and the container interfaces that host them. The module interface specifies the interface to a module, which is used by the container to call module operations. The container interface specifies the functions that the container provides for a module. Different bindings provide mapping for programming languages. Currently three language bindings are available: C, C++ and Ada.

In conclusion, there are some advantages and disadvantages in ECOA. The advantages are the fact that this standard allow the developer the option to choose pre-existing schemes, but he can also create new ones, which he can adapt to his needs. This standard also allows the developer to choose between different programming languages like C, C++ and Ada as previously stated. One of the disadvantages is the fact that this is a relatively new standard and because of that there can be some errors. However, there are frequent updates to fight these issues. Also, it was created based on military systems, as it may have some doctrine that does not suit civilian tasks. Finally, it is designed for UAVs, which does not provide interoperability when in a joint operations scenario with different types of vehicles, such as UGVs or UUVs.

3.1.8. Battle Management Language (BML)

BML was created in 2001 by the U.S. Army. It is an XML based data model, with the aim of providing a data model to exchange military orders, command and control reports and requests, between military forces, manned or unmanned [176]. From the onset, it was designed to allow interoperability between manned and robotic forces, providing a clear, unambiguous, machine-readable syntax that can be used in a military environment. If all orders and reports are provided in BML, it is much easier to obtain a situational awareness tool that integrates multiple units, vehicles, and systems. BML also provides a good and realistic way of conducting simulations, both in an entirely simulated environment, and in a mixed reality environment.

BML was created by Simulation Interoperability Standards Organization (SISO) study group. Originally it was based on the *Command and Control Simulation Interface Language*, which is a data model used to simulate small units and platforms, but it is not consistent with the evolution of C2 data, and therefore it was not maintained as standard and evolved into BML[177].

The implementation of BML uses the *Joint Consultation, Command and Control Information Exchange Data Model* (JC3IEDM) as a system-independent common vocabulary for passing plans, orders, and reports among C2 systems and simulations. BML enables interoperability amongst services, allowing the use of joint and coalition systems by providing a common means of exchanging information that all C2 and simulation systems can implement[178], as is represented in Figure 3-10.

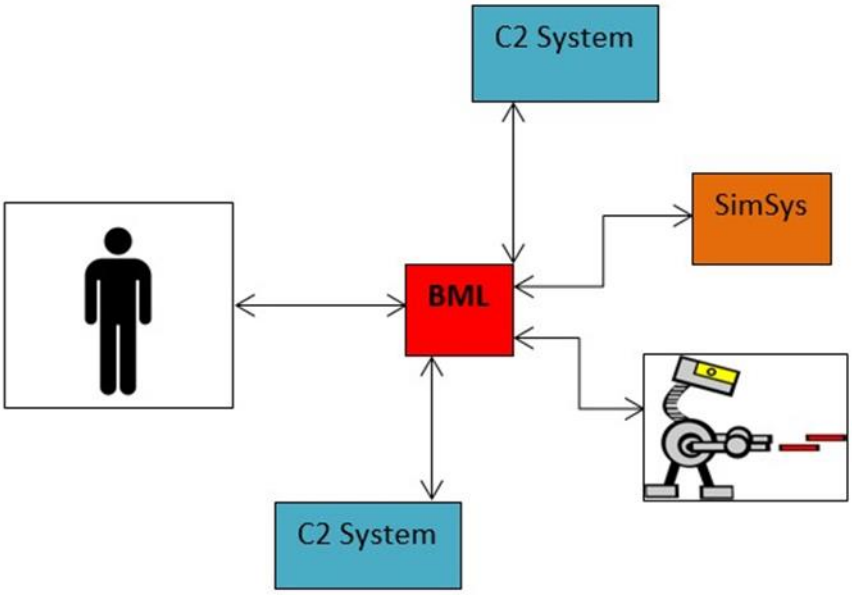

Figure 3-10 - BML Connection

Integrating human units, robotic units and simulated units, with various C2 Systems

The main characteristics of BML are:

- It is expressive and precise, because it is based in formal rules;
- It is machine-readable, as the military information can be validated and processed based on a reference model;
- It is understandable by humans, because the expressions are basically in English, and it is designed to support multiple military doctrines, including NATO's;
- It is multi-domain, since it can be used in air, maritime, land and joint environments;
- It is independent of the information exchange mechanism, since it is a data model and can use any communication infrastructure (including paper messages);
- It is an international data model.

BML grammar is based on English vocabulary and on military specific definitions and, as previously stated, it uses missions listed in the JC3IEDM database, creating plans to help executing those tasks. Thus, one may question the utility of BML given that there already is JC3IEDM. The answer for that is the fact that in JC3IEDM the database depends on human operators to characterize the type of the mission and to assign tasks. On the contrary, in BML, there is a grammar that forms unambiguous expressions to communicate those orders to a person, a UxS, or some other system[179].

The BML data model is a based on a *5W*s concept: *Who*, *What*, *Where*, *When* and *Why*. Based on these 5 questions, the grammar defines that each sentence addresses one *W word* and gives the answer for it, which can be a sequence of actions based on the JC3IEDM database. Searching for answers in the database can be difficult if there is no method to do it.

Scalability and robustness of the BML infrastructure is critical. BML message validation and error handling are important capabilities to ensure robustness. Multithreading and load balancing further increases server scalability[180].

In conclusion, the advantages of BML are the fact that it is a digital interface that promotes UxS flexibility, integrating them not only with other UxS and C2 systems, but also with human forces. Therefore, management of resources employed in a mission may be planned and allocated dynamically with this interface, making the operators work easier. As for the disadvantages, one of them is the fact that it is only planned for UGVs and UAVs. Thus, it is not used in the maritime environment, and it cannot be applied in joint forces with USVs or UUVs. Also, because of underlying doctrine for which it was designed, it is used only by the military, and it is not appropriate for civilian applications. Finally, BML only defines messages addressed to the assets as a military force, and do not assume or define any internal UxS architecture.

An example of how traditional military orders are expressed in BML can be obtained, e.g, in [181], which describes an exercise actually carried out by NATO forces in the Netherlands. In plain English, using military terms, the mission could be stated as:

MNC Commander's Intent. My intent is to direct two-division movement from Tactical Assembly Area (TAA) to blocking positions along PL TULIP. In the event of incursion by BRADYLAND forces, MNC forces will not allow their progress north of the buffer zone. Keys to success include safe arrival at PL TULIP, construct and occupy blocking positions along PL TULIP, to prohibit the advance of enemy forces beyond the northern boundary of the buffer zone. The main effort is the counterattacks to eject the

BRADYLAND forces from GENERICLAND and restore the international border. The end state is achieved when the UN recognized border between BRADYLAND and GENERICLAND is re-established.

The BML code generated and sent to units would be:

[Expanded Purpose]
Status-Report neg hostile position combat-unit at BUFFER ZONE at TP6 RPTFCT label-ep-a;
Task-Report establish MNC "stabilized area" at GENERICLAND start at TP6 RPTFCT label-ep-b;
[Key Tasks]
move MNC OPEN from TAA to PL TULIP start at TP4 in-order-to enable label-kt-b label-kt-a;
occupy MNC OPEN combat zone at BUFFER ZONE start nlt TP5 in-order-to enable label-es-a label-kt-b;
counterattack MNC OPEN Enemy at BUFFER ZONE start nlt TP5 in-order-to enable label-es-a label-kt-c;
[End State]
Task-Report establish MNC border at "UN Recognized Border" end nlt TP6 RPTFCT in-order-to secure label-es-a;

3.1.9. **Autonomous Vehicle Command Language (AVCL)**

AVCL was developed by the Naval Postgraduate School (NPS) in 2005 and it is a data model based on XML. It is used to define UxS tasking, inter vehicle communication and mission results[182]. AVCL also defines datatypes that should be used when exchanging information between vehicles.

The advantage of converting the UxV ontology into this data model can be the use of planning development and analysis tools for arbitrary vehicles. Due to this, different vehicles can exchange data, enabling interoperability.

The AVCL data model is divided into three parts: mission preparation, communication and mission results. *Mission preparation*'s objective is to define the mission requirements [183]. *Communication*'s function is to define the format of the messages exchanged with other UxS or GCSs. *Mission results* is used to record and pass to the Ground Control Station data such as telemetry and control orders, or contacts and messages sent or received.

Mission's requirements have two ways to be specified for the vehicle. Firstly, it can be a sequence of task-level script commands. This means it will have a list of commands, and each one is executed at a time, in the predetermined order. For example, the simple task could be the request for a vehicle state parameter, and more complex one could be giving waypoints to navigate. Secondly, it can be a set of goals and constraints for the mission. This type of specification is required for more complex missions and vehicles.

The AVCL schema defines several string-based enumerations, for example, when reporting sensor type. The use of meaningful strings, instead of integers makes for more readable and intuitive documents. AVCL also supports reusable data types with more complex structures, including attribute names and types, and inheritance or inclusion mechanisms (in the computer science sense of the word).

One of simplest types of data is *noValueElementType*. This type has no child elements, and it is created simply by assigning an element name (and no more operations can be done on it). The slightly more complex data type *scalarElementType*, that corresponds do the classical scalar type, where an element has a name and a value, that can be changed during execution[182]. To specify geographic positions has two types: xyElementType, which encodes the position in a cartesian coordinate pair; and latitudeLongitudeElementType, which encodes the position in latitude and longitude. As types get more complex, they also become more powerful, and the whole mission of a UxS can be encoded in an element (named rootElement) that has many attributes and child elements.

This data model also specifies behavior scripts, used to complete one or more task level behaviors. These behaviors are divided according to the vehicle type, as they operate in different environments. AVCL presents 30 UUV behaviors, such as the CompositeWaypoint, which has parameters such as depth. Not all these behaviors make sense for UGV, USV and UAV, so these have mainly subsets of the behaviors defined of the UUV.

To conclude, AVCL is a data model that allows interoperability between different vehicles, using a common data model to plan missions and share results. It does not depend on specific hardware and can be used for heterogeneous fleets of UxS.

3.1.10. Multi-sensor Aerospace-ground Joint Intelligence surveillance and reconnaissance Interoperability Coalition (MAJIIC)

MAJIIC is a project started in 2006. MAJIIC is a multi-national effort to enable interoperability between NATO and national ISR and C2 systems using common interfaces for data formats. Working with nine nations under a Memorandum of Understanding (MOU), its aim is to improve commander's situation awareness by developing and evaluating operational and technical means for ISR assets interoperability in a coalition environment. MAJIIC has since created an interface based on STANAG 4559 (NATO Standard ISR Library Interface) for metadata-based access to archive data from any Coalition Shared Database (CSD) in the MAJIIC environment. With the development of the CSD and CONOPs for coalition ISR operations, MAJIIC also provides a means for the U.S. DoD, intelligence and coalition communities to assess new ISR net-centric data sharing concepts and solutions[184].

In order to achieve this improvement, MAJIIC is divided into three primary perspectives:

- The operational perspective includes development and demonstration of concepts of employment and tactics, techniques and procedures for collaborative employment and use of coalition ISR assets in support of military missions;
- The architectural view includes development of procedures and technology for sharing ISR data and information, system data model design principles, tools and technology for collaboration, and tools for managing coalition ISR assets;
- The technical point of view includes definition and development of key data model for the various sensor and data types, tools to support common geo-registration, and data exploitation.

MAJIIC has some specifications that are inherent to its military origin. In order to approach interoperability, it addresses the exchange of data from various ISR sensors in a network-enabled manner. Thus, it is guided by operational doctrine, based in providing a detailed description of how a system is employed, including resources and capacities, information operations techniques, tactics and procedures, and other standards and guidelines. This operational expertise is achieved with the cooperation of all nations involved and with NATO multinational and national activities and programs.

MAJIIC addresses a wide range of needs from those of small tactical commands to those of highly capable multi-user systems, being a flexible and wide-reaching project. Although originally developed for UAVs, it also addresses UGVs and USVs. As is shown in Figure 3-11, there are several types of sensors used in MAJIIC: Ground Moving Target Indicator (GMTI), which is a land RADAR; SAR, which is used to create images of determined objects; Electro-optical (EO) and infra-red (IR) imaging and video sensor; Electronic Warfare Support Measures (ESM) sensors, because it is a military project; artillery locating RADAR.

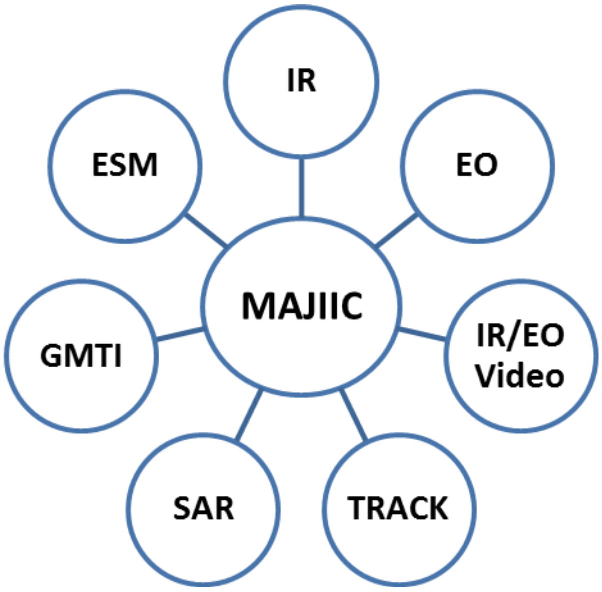

Figure 3-11 - MAJIIC Data exchange

Each system should provide data to a ground station or another component that is inside a common network structure. This exchange should be based on STANAGs, which include: STANAG 4545, for EO, IR and SAR still imagery; STANAG 4607, for GMTI data; STANAG 4609 for EO and IR motion imagery (video); and STANAG 5516, for track and management messages. In order to achieve this data exchange, MAJIIC has implemented an interface based on STANAG 4559 (NATO Standard ISR Library Interface). This will allow metadata-based access from any coalition shared databased throughout the MAJIIC environment[185].

In conclusion, this is a military project, with the advantages of having interoperability between several systems, either in land, sea or air and therefore, better decision-making capacities for the commanders of the forces. Another advantage is the fact that it is adaptable to any network type of bandwidth, because this may be a limitation in real time situations. This allows MAJIIC to be a data model with a wide variety of users and may be used in different scenarios. However, MAJIIC also presents some disadvantages. One of them is the fact that it is a military based data model, as its doctrine is focused on the Armed Forces. Thus, it is not appropriate for civilian tasks. Also, it does not address specific issues of UxS themselves (only their payload), as it only provides the set of messages and data formats for sensors that should be implemented to have compliancy with these interfaces.

3.1.11. **NATO Industrial Advisory Group (NIAG) Subgroup 157 (Study on Multi-Domain Unmanned Vehicle Control) (NIAG - 157)**

NIAG-157 defines a data model for a Multi-Domain Control System (MDCS) and was created in 2011 by a NATO working group. The objective is to enable a NATO interoperable control system for UxVs whether operating in air, sea or ground environments.

The requirements are that the data model should:

- Be compatible with other open data models or components;
- Provide an open system interface with external systems;
- Be capable to support changing missions;
- Support rapid integration of new unmanned platforms and its subsystems;
- Separate safety of flight (or equivalent) from mission support operations;
- Define architectural requirements relating to security and information assurance.

This data model is organized in four layers: application, platform, adapt and physical layer (Figure 3-12). It also defines a Logical Data Model (LDM), used throughout the system. A full definition of this LDM can be found in [18] , where it is defined in Unified Modelling Language (UML). This LDM contains many types of data that cover a very broad set of concepts.

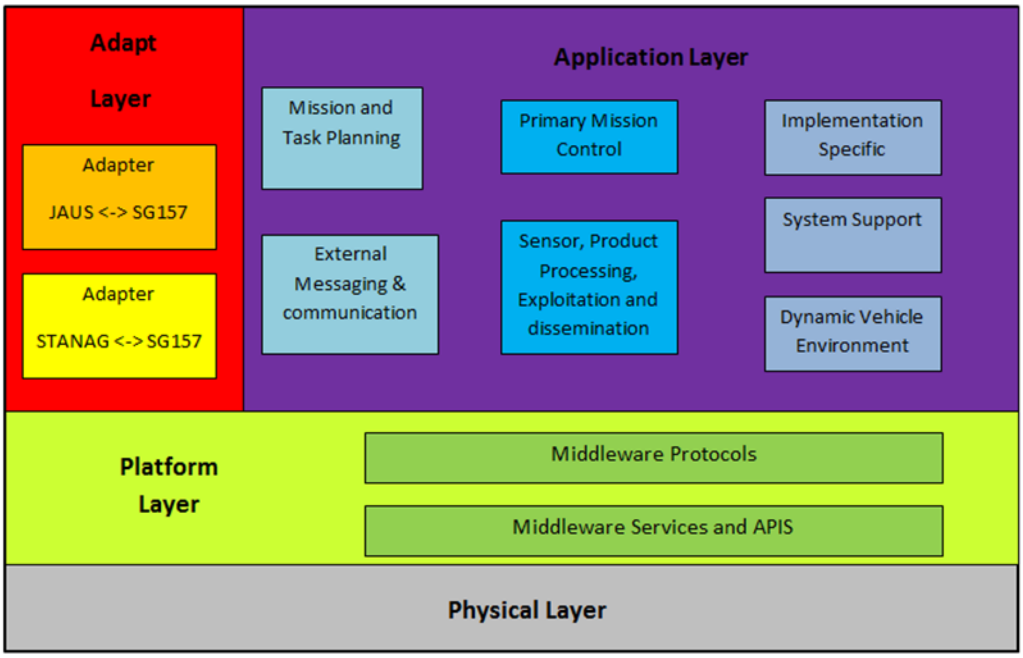

Figure 3-12 - NIAG 157 Layers

The *Physical Layer* provides the interface with the control station hardware that communicates with the vehicle.

The *Platform Layer* manages the control station, providing services to the application and adapt layers, and sending data to be transmitted by the physical layer when necessary. It includes middleware services and APIs that can be accessed by the other layers. It has its own middleware protocols to ensure information passing across the APIs.

The *Adapt layer* allows interoperability of systems compatible with NIAG-157 (that use its LDM) and systems designed to operate using external standards, for example STANAG 4586 and JAUS. It is basically composed of modular translation libraries that pass information to and from NIAG-157's LDM to whatever data model legacy systems use.

The adapter layer supports interoperability with other systems including STANAG 4586 Compliant platforms or JAUS.

The *Application Layer* provides the core of NIAG-157's control station functionality.

To improve maintainability and management, the Application layer is partitioned into application domains based on subject matter expertise:

- Primary mission control;
- Mission and task planning;
- Sensor product processing, exploitation and dissemination;
- External messaging and communication;
- System support;
- Dynamic vehicle environment;
- Implementation specific functions;

Primary mission control covers the key activities of the vehicle during the operation like checking the objectives and managing communication with the control station and other UxVs.

Mission and task planning covers the sensor data usage during and after the mission. It manages the route to the objectives according to the data collected about the environment and battlespace.

Sensor product processing, exploitation and dissemination manages the archive of the data that is collected by the sensors and is responsible to send it via the C4I interfaces.

External messaging and communication provides tactical messaging capability and collaboration tool capability to external communication being performed during all the phases of the mission.

System support covers activities related to the maintenance of the MDCS itself, providing support tools, training capability and administrative tools.

Dynamic vehicle environment covers the issues related to the environment where the UxV is operating, proving the necessary situational awareness, including interactions with other vehicles in the battlespace, collision avoidance, weather and terrain issues, rule compliance, etc.

Implementation specific functions covers the human machine interface for operational and maintenance phases. Since there are NATO standards for human machine interfaces for UAV control stations, these are left out of NIAG-157.

In conclusion, NIAG -157 provides a multi-domain model for control stations, and thus provided UxS interoperability by allowing the same control station to interact with different vehicles. Its data model (LDM) is modular, allowing incremental improvements at a system, subsystem or component level. It supports multiple command and reporting standards to communicate with the UxV, and although having a string emphasis on UAVs, takes into account the characteristics of the other UxVs. It was designed to be *future proof* in the sense that it is very modular and tries to separate clearly different functions that a GCS should have. However, it only covers the ground segment of the UxS, it is NATO initiative (although other nations can have access to it), and it hasn't had much success amongst the research community.

A follow-up on NIAG 157 was the NATO Industrial Advisory Group (NIAG) Subgroup 202 (Study on development of conceptual data model for a multi - domain unmanned platform control system) (NIAG - 202). This group lasted from 2015 to the end of 2016 and according to the group's documentation, "*The aim of this Study Group is to develop a data model that would represent all the information required for a Control System to operate assets from multiple domains, and to develop draft guidance on how to implement and test the system. A secondary objective is to propose a plan for NATO development of a prototype*"

The final report, which has a "NATO Unclassified" security classification and is accessible to NATO countries, NATO partnership for peace, Australia and Israel[186] is a large document with many recommendations, analysis of requirements, conceptual descriptions, test criteria, etc., but falls short of defining or adopting an actual protocol.

3.1.12. **Robot Operating System (ROS)**

ROS was originally developed in 2007 by the Stanford Artificial Intelligence Laboratory (SAIL) with the support of the Stanford AI Robot project. ROS is an open-source, framework for robot application development maintained by the Open Source Robotics Foundation (OSRF). A ROS system is comprised of several independent nodes that communicate with each other using a publish / subscribe messaging model that can be deployed over different computers[187].

The purpose of this system is to facilitate the creation of new applications for robots, by exploiting libraries, algorithms, and hardware components. Its principal objective is to maximize the reusability of already available robot sensor visualizations, sensor fusion and control algorithms. ROS is a node-based architecture which allows the system to be flexible and easily reconfigurable[19].

Concretely, it helps developers by providing hardware abstraction, device drivers, libraries, visualizers, message-passing and a package management system. ROS is licensed under an open source, Berkeley Software Distribution (BSD) license[188].

ROS provides a heterogeneous computing cluster and structured communication layer above the host operating systems. It was designed based on a modular tools-based philosophy for software development. The large developer base means it became a de-facto standard framework for robotic platforms[189]. From our experience, it is so widespread that most development groups use ROS in some fashion.

ROS defines message types for the common use of robot sensor data such as images, inertial measurements, GPS and odometer data. Each sensor of data processor is known as a “node”, that may communicate with the “ROS Master”, which controls the whole system, or directly other nodes, as shown in Figure 3-13.

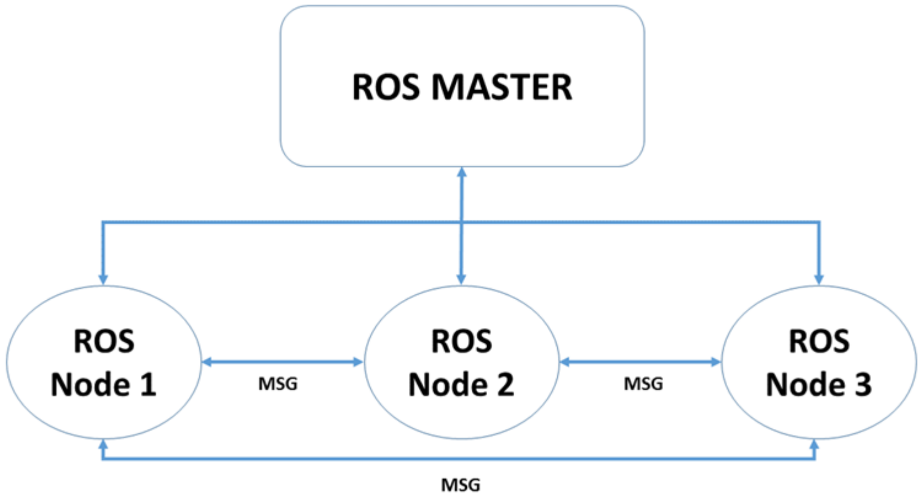

Figure 3-13 - Basic ROS functional system

Example of a basic ROS functional system, with a master and several notes sending messages to each other.

Thus, it's unnecessary to explicitly define separate data structures for integrating different components. However, these messages have been created on demand and they are continuously evolving as new needs are identified.

A ROS system is divided in basic concepts: nodes, messages (MSG), topics and services.

As previously stated, ROS is a modular framework and nodes are the different processing modules. Because of this we can describe the system with a graph, where each node is a module. For example, one node can control the engine of a vehicle, or it can be responsible for its location, or for planning a navigation route[190].

MSG are the method how nodes communicate. They are a data structures with various fields.

Topics are unique identifiers that represent communication channels, each targeted at a specific type of subject. MSG are routed in topics by a TCP/IP transport system. Nodes send/receive MSG by publishing/subscribing to a determined topic. If a module is interested in receiving information present in a topic, it simply subscribes to the corresponding topic. Each node can publish or subscribe different topics and topics can have multiple publishers or subscribers (many-to-many relationship). However, publishers and subscribers are not conscious of each other's presence.

Services are a different communication paradigm. They implement a synchronous data exchange mechanism (e.g. server-client model). They have a string which represent its name (similar to a ROS Topic) and two MSGs, one for the request and another for the service response.

There are various tools that can be used with ROS. One of those is debugging a single node. This is a consequence of ROS being a modular framework. Without this capacity, the system could not do reset to only one node. For example, to do a reset in the camera elements it would be necessary to reset all the nodes that are related, like the pose detector or the object recognizer. With ROS this is minimized because with a modular framework the graph becomes dynamic and there is the possibility to reset only the necessary node.

The logging and playback functionality is important to simplify and make the system more efficient, and mainly for debugging. Every MSG in ROS can be saved in memory (usually to disk in what is known as a "bag") to be later replayed, so this framework gives the opportunity to play back messages that can be used in the same nodes or even in others (if they require the same function). This may be used to find bugs, even in complex asynchronous systems, or to test new components in a realistic but simulated and reproducible environment.

There are several visualization tools in ROS, that allow the programmer to have a dynamic vision of the ROS graph. This can be used to better understand what is going on, test modifications, and make the system more efficient.

One of the main advantages of ROS, when compared to other frameworks, is its peer-to-peer network topology. A central server, that would know all about all nodes and distribute the MSG for all the other hosts, would require a lot of computing power and communication bandwidth, especially in large and complex systems. It also eliminates a very crucial single failure point. The second main advantage of ROS is the fact that it can be used with different programming languages, such as C++, Python, Octave or LISP, giving the programmer the capacity to choose the one most suitable for the application. Another advantage is the fact that it is free and Open-Source. Any person can contribute with libraries, and this gives the opportunity of having a variety of designs and complex systems[19].

On the other hand, ROS also has some disadvantages. One of them is the overhead of the messaging system, which isn't as compact as other systems, and that can be a problem in large systems with many topics and services. Another disadvantage cited by the ROS community, is the fact that it is a difficult system to get familiarized with. Finally, ROS may not be the best choice for multiple robot teams, as currently there is no standard way to build them, and in most cases, for simplicity, each robot acts as a structure bellow a master ROS.

In conclusion, ROS has a variety of advantages comparing to other frameworks. The main ones are the fact that this is a modular system, composed by nodes, and nodes can easily be changed if necessary, or switched without major changes or compatibility problems. It is also an open-source system, which allows the programmer to have various packages available that were developed by other researchers.

We will now present an example of ROS code.

Let's assume that we have a ROS node, a robot controller, that controls the locomotion of a robot by subscribing to *Twist* messages on the '*/controller/command*' ROS Topic. The *Twist* data type has two *Vector3* fields: three-dimensional linear (*x*, *y* and *z*) and angular velocities (also labeled *x*, *y* and *z*).

Nonzero entries in the *x* and *y* fields of linear velocity causes the robot to move forwards and backwards (x), or strafe left and right (y), in the robot's base odometry frame. A nonzero entry in the *z* field of angular velocity causes the robot to turn (yaw). A single command will only move the robot for a short period of time before stopping, so it does not run off into the wall (or you) when commands stop coming for any reason. Velocities are in units of *m/s* and *rad/s*.

In the code below, we will rotate and move left.

```
#include <ros/ros.h>
#include <geometry_msgs/Twist.h>

int main(int argc, char** argv)
{
  //init the ROS node
```

```
    ros::init(argc, argv, "driver");
    ros::NodeHandle nh;

    //set up the publisher for the cmd_vel topic, that will publish a "Twist" structure
    ros::Publisher  cmd_vel_pub_= nh.advertise<geometry_msgs::Twist>("/controller/command", 1);

    //we will be sending commands of type "Twist", so we create the object "cmd"
    geometry_msgs::Twist cmd;

    //prepare to turn left (yaw) and drive forward at the same time
    cmd.angular.z = 0.75;
    cmd.linear.x = 0.25;

    //publish the assembled command, that will be executed by a mode that subscribes "cmd_vel_pub"
    cmd_vel_pub.publish(cmd);

  (...)

  }
```

3.1.13. **Lightweight Communications and Marshaling (LCM)**

LCM was developed in 2006 at MIT. It is a low-latency, high-throughput communications framework that scales to many senders and receivers. LCM consists in a system whose objective is message passing and data marshalling in real-time, to solve the interprocess communication problem (communication between modules that form an autonomous system). It provides a publish/subscribe message format and XDRstyle (XML Data Reduced) message specification language, but it also has connections for applications in C, Java and Python. It uses the User Datagram Protocol (UDP), which is a communication protocol of the transport layer, for message exchanging, which is highly scalable, and is a good choice for real-time communications[191].

In this context, data marshalling is LCM's ability to encode and decode structured data into a binary stream that can be transmitted in a UDP packet over the network, using its standard libraries.

LCM defines several data types, independent of the platform and represented as a byte stream[192],[193], and the processes that wish to communicate using LCM should previously agree on the data type format that will be used to exchange data.

The communications aspect of LCM can be summarized as a publish-subscribe based messaging system that uses UDP multicast as its underlying transport layer. Under the publish-subscribe model, each message is transmitted on a named channel, and modules subscribe to the channels required to complete their designated tasks. It is typically the case (though not enforced by LCM) that all the messages on a channel are of a single pre-specified type[194].

To assist development of modular software systems, LCM provides several tools useful for logging, replaying, and inspecting traffic. The logging tools are like those found in many interprocess communications systems and allow LCM traffic to be recorded to a file for playback or analysis at a later point in time. The inspection tools allow real-time decoding and display of LCM traffic with no system overhead (such as additional network bandwidth) or developer effort. Together, these tools allow a developer to rapidly and efficiently analyze the behavior and performance of an LCM system[195],[196].

Similarly, to ROS, LCM, using only multicast UDP messages, avoids a centralized communication hub. Also, like ROS, it has a powerful tool for debugging and inspecting transmitted messages[197].

On the other hand, LCM also presents some disadvantages. One of them is the fact that it is not ready to use with different types of vehicles, such as UAVs or UUVs. LCM has already been tested with UAVs and UUVs and results were positive[192], but developers must adapt the framework to this reality, and this adaptation is not standard. Also, it does not provide an underlying UxS architecture, as some standards do. Instead, it presents a framework only for communication between modules, which can be a problem, depending on the type of project.

3.1.14. **Micro Aerial Vehicle Communication protocol (MAVlink)**

MAVlink is a micro air vehicle (MAV) marshalling and communications library specially focused on MAVs and it was developed in 2009 by Lornez Meier at the ETH Zürich. MAVlink is a protocol for lightweight communication between Micro Air Vehicles (or a warm of them) and/or Ground Control Stations (GCS). It serializes C-structs for serial channels and can be used with any type of radio modem. Message definitions are created in XML, and then converted into C header files. MAVlink is also used for Linux inter-process and ground link communication in several software packages (ROS, APM planner)[198],[199],[200].

MAVLink acts like a mechanism with a wide non-filtered broadcast of messages that each component or sub-system can receive and read. Complementarily, every component can broadcast messages. The messages have a double Cyclic Redundancy Check (CRC) correction process with an extra byte for the second checksum. That improves the consistence of communications and the data package contents[201].

MAVLink packets are composed by a header, message and CRC correction. In the header section, there is a frame identifier, the message length, packet sequence number, system ID of the sending system (because there can be various vehicles), component ID of the sending system (it specifies the actual component of the vehicle) and the ID of the message. Message formats may vary, depending on the type of message, but usually in all autopilots there are *heartbeat*, *command* and *waypoint* management messages, although this depends on the autopilot that is being used[202],[203]. As previously said, CRC is used to confirm that the message is correct.

The protocol is supported by an assortment of autopilots and ground control software including Ardupilot, Parrot AR, Pixhawk, QGroundControl, APM Planner, and more. By utilizing this protocol, the payload firmware can seamlessly interface with a wide variety of existing autopilot systems[204],[205].

One of the advantages of this protocol is the easy access to common data, including messages, tutorials for the integration or even for the message formats. This occurs because it is a GNU - Lesser General Public License (LGPL) licensed protocol, which is a free software license. Another advantage is the possibility to create new messages that may not exist already, because of a specific mission requirement. Finally, it is a lightweight protocol, as it provides messages with a small header, turning the process very fast and efficient. On the other hand, some disadvantages of this protocol are the fact that it is specific for air vehicles, and not for other types. Also, it is a simple library, as it is mostly used for civilian applications, because complex scenarios, such as military missions, require specific and complex sets of messages[206].

In conclusion, MAVLink represents a simple and lightweight protocol, ideal for MAVs. The easy access to applications and libraries make it a very good option for any researcher that wants to develop their software and simulate the UxS on a computer. As it is widely used open-source system, it also has very good support. However, as it is a simple protocol, it may not have the characteristics that are needed for complex tasks, like military ones.

We shall now provide an example of MAVLink Code.

In the follow example, we use the C++ mavros package library in a ROS node that must position a UAV at an altitude of 20 meters, at latitude of 20º and longitude of 10º.

```
(…)
// create a ROS service client
ros::ServiceClient client = nh.serviceClient<mavros_msgs::Way-
pointPush>("topic_name");
mavros_msgs::WaypointPush srv;
mavros_msgs::Waypoint wp;
//create waypoint message structure
wp.frame = mavros_msgs::Waypoint::FRAME_GLOBAL;
wp.command = mavros_msgs::ComandCode::NAV_WAYPOINT;
wp.is_current = false;
wp.x_lat = 20;
wp.y_long = 10;
wp.z_alt = 20;
//push the waypoint data to the service variable
srv.request.waypoints.push_back(wp);
// call the MAVROS service to send the data to the UAV's autopilot in the MAVLink for-
mat
client.call(srv);
(…)
```

3.1.15. Inter-Module Communication (IMC)

IMC protocol was designed and implemented in the Underwater Systems and Technology Laboratory (LSTS) of the Engineering School of Oporto University, Portugal, in 2009. IMC is a message-oriented protocol that defines a common control message set which was created to be understood by all types of vehicles and computer nodes. It is based on a message passing concept. These messages are divided in groups, in a modular way, providing different control and sensing layers[163].

One of the objectives of this protocol is to have hardware abstraction, which means that it can be used with different hardware components. All the messages can be serialized. This protocol does not assume a specific software architecture for client applications, contrasting in this with most other protocols.

The set of control messages that IMC provides can be divided into several logical groups, for networked vehicles and sensor operations. Mission control messages define the type of mission and its life-cycle. It is used for the interface between a Command and Control Unit (CCU) and a mission supervisor module. *Vehicle control* messages are used to control the vehicle from an external source, giving commands, for example maneuver requests, or checking its state. *Maneuver* messages set maneuvers, which have specific commands and execution states associated. The simpler are waypoint tracking maneuvers, for example, to go from one point to another. *Guidance* messages define guidance characteristics used in the maneuvers. These maneuvers are done autonomously, so the vehicle must receive some parameters, such as heading, depth or velocity. *Navigation* messages report the navigation state of the vehicle. *Sensor* messages report sensors state, by checking the readings of the hardware controllers. This reading can be, for example, a GPS, an IMU, among others. Finally, *actuator* messages specify the interface with hardware controllers, based on the previous messages and on the requirements, they need[163]. An example of the IMC message flow is illustrated in Figure 3-14.

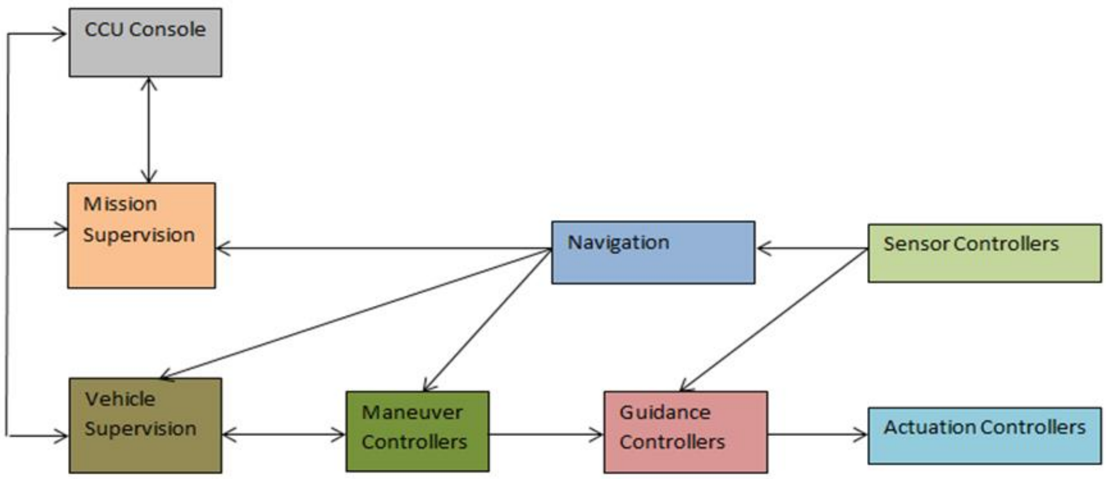

Figure 3-14 - IMC Message Flow

As previously stated, IMC is a modular protocol. Each component can run its software in logical isolation, because the exchange of messages is done only by the IMC protocol, and a simulator can replace the other components and the physical environment[207]. This exchange of data is done using a message bus abstraction and provides transport mechanisms for external communications. It keeps data integrity by having a check sum field, using CRC-16.

In conclusion, IMC is a modular protocol, designed with various types of UxV in mind, and supporting different types of hardware. Also, it allows low and high-level commands, in order to have generic messages and also more specific ones. As for disadvantages, there are not many vehicles that operate with this protocol.

3.2. **Comparisons**

This chapter aims to compare the IBBs that were referred in the previous chapter.

3.2.1. **Summary of advantages and disadvantages of the IBBs reviewed.**

The advantages and disadvantages of the IBBs reviewed in the previous chapter can be summarized in Table 3-1.

Table 3-1 - Advantage and disadvantages of some IBBs

Communications Methods	Advantages	Disadvantages
STANAG 4586	• Interoperability between NATO systems;	• Is not addressed to an operational improvement; • Only for UAVs.
JAUS	• Adaptable to any type of UxS;	• Limits the creativity of the developer;
	• Independent from almost all external variables;	• It's necessary to have knowledge on the assignment of subsystems and nodes IDs to address messages.
	• It's modular.	
MOOS	• Simple even with many participating clients; • The client is independent from other connections between the server and other clients; • Has wide support with the research community.	• Centralized topology makes him vulnerable to "bottle-necking";
CompactCL	• Low bandwidth;	• Designed to use only in UUVs;
	• C structure messages allow the developer to choose other similar constructs;	• Requires the use of WHOI micro modems.
	• It has a wide set of options to be implemented.	
CommonCL	• It's a language that coordinates other existing architectures in the same environment;	• Designed to use only in UUVs;
	• This language provides methods to develop a multi robot system in an underwater environment.	• Multi robot cooperation is under development and it can still be improved.
CLARAty	• Architecture for generic and reusable robotic components; • Provides easy introducing of new technologies into the robotic system.	• Designed to use only in UGVs;
ECOA	• Adaptable to new needs;	• New architecture and because of that there can be some errors;
	• The developer can choose different programming languages.	• Based on military systems so it may have some doctrine that does not suit civilian tasks;
		• Designed to use only in UAVs.

BML	• It's a digital interface that promotes UGV and UAV's systems flexibility; • C2 integrated systems helps achieving interoperability;	• Designed to use only in UAVs and UGVs; • Not appropriate for civilian applications. • Only defines messages and the doctrine applied to them.
AVCL	• By having a data model, task success improves; • Allows the developer to choose the data model that suits the mission or the environment in a better way.	• It needs an adapter to do the conversion from another language to this one.
MAJIIC	• Interoperability between several systems; • Adaptable to any network type of bandwidth.	• Not appropriate for civilian tasks; • Only provides the set of messages and data formats.
NIAGSG-157	• It supports the acquisition of segments at system, subsystem, component or service • It supports other open architecture's standards; • It is scalable, from small to large systems, in matters of computational power; • It separates safety of flight from mission support operations; • It was designed to assume evolutions of different nature;	• It's only for NATO country members; • It's focused on the control station and not on the vehicles; • There is the need of having adapters for other standards. • Does not has an easy support for the research community when comparing with other languages or standards.
ROS	• It's a reliable framework composed by nodes; • It's an open-source system; • Peer-to-Peer topology; • It can be used in different languages.	• Overhead of the messaging system can accumulate in large systems; • It's a difficult system to get familiarized with; • There is no standard way to build the robots.
LCM	• It uses UDP protocol; • It's capable of being integrated into a variety of platforms and operating systems; • Tool for easy debug and inspection of messages.	• It is not ready to use with different types of vehicles; • It does not provide the UxS architecture.
MAVLink	• Easy access to common data; • The possibility to create new messages that may not exist already.	• It is specific for air vehicles and not for other types; • Its libraries are simple, requiring specific and complex sets of messages for more complex scenarios.
IMC	• It's a modular protocol; • Native support can be generated for different programming languages; • Allows low and high-level commands.	• Support is not easy to access; • There are not many vehicles that operate with this protocol.

3.2.2. Comparison of the main characteristics of the IBBs reviewed

The main characteristics of each IBB can be summarized in the following table:

Table 3-2 - Characteristics of the IBBs reviewed

Communications Method	Model Type	Responsible Organization/Person	Vehicle type	Accessiblity	Importance
STANAG 4586	Standard	NATO	UAV	Open	++
JAUS	Standard	Society of Automotive Engineers	UxV	Open	+++
MOOS	Standard	Mobile Robotics Group	UGV, UUV and USV	Open	++
CompactCL	Standard	Office of Naval Research	UUV and USV	Open	+
CommonCL	Data Model	Autonomous Undersea Systems Institute	UUV and USV	Open	+
CLARAty	Standard	Jet Propulsion Laboratory in colaboration with various universities	UGV	-	+
ECOA	Standard	Direction Générale de l'Armement & Defence Science and Technology Laboratory	UAV	Military	+
BML	Data Model	Simulation Interoperability Standards Organization	UAV and UGV	Military	+
MAJIIC	Data Model	NATO	UxV	Military	+
AVCL	Data Model	Naval Postgraduate School	UxV	Open	+
NIAGSG-157	Data Model	NATO Industrial Advisory Group	UxV	Military	++
MIP	Data Model	National Command and Control Information Systems	UxV	Open	+
ROS	Framework	Open Source Robotics Fundation	UxV	BSD License	+++
LCM	Framework	MIT DARPA Urban Challenge Team	UGV	Open	+
MAVLink	Protocol	Lorenz Meier	UAV	LGPL License	+++
IMC	Protocol	Underwater Systems and Technology Laboratory	UxV	Open	+

Model type is the first parameter that is specified in Table 3-2. This is the parameter that divides each system according to its purpose, as there are some that specify the whole architecture, while there are others that are more specific, focusing only on information exchange between UxS. Therefore, four main types are proposed: standard, framework, protocol and data model.

In this classification, a *standard* (already defined in chapter 2) can be simply defined as a set of rules and models that a system should have, and it is used here as a broad concept. Therefore, IBBs classified as standards are those that specify a general architecture, which can be for the whole system or for communications. An obvious example of this is STANAG 4586, which is a standard that specifies the whole UAV, amongst much more information. Another example is CompactCL, which is a standard that specifies an architecture for inter UUV and UUV- human communication.

The second type is *data model* (also defined in chapter 2), which is an abstract way of describing how data is represented in the communication system. It aims to conceptualize and structure the communication layer. Therefore, one example of data model is BML, which specifies and conceptualizes doctrine that should be used in UxS. Another example is AVCL, as it defines data types that should be used to exchange information.

Another type is framework. A framework (also defined in chapter 2) can be defined as a support structure (software) intended to guide the building of something. In this case, it supports the creation of a certain unmanned system or communications architecture. One main example of this is ROS, which is a framework that provides tools in order to develop a whole system, in this case, with the creation of nodes. Another example of a framework is LCM, which has tools and applications in order to help its message passing communication system.

Finally, the last type is *protocol* (also defined in chapter 2) and it can be defined as a set of regulations that determine how the data should be transmitted over the network. While a standard is a broader concept, a protocol can be seen as a more specific one, which only specifies message exchange. The main example of this type is MAVLink. This protocol specifies the message format that UAVs must use in order to exchange commands and information. The other case is IMC, and it also defines common messages that should be exchanged between systems.

The second parameter in Table 3-2 is the responsible organization or person. This is an important parameter, not only because it gives the idea of how and why it was created, but also because this is the way of getting help if something is needed in the implementation of the architecture. There are some developers that are from military organizations, like NATO and many of their advisor groups, because the UxV field is very important in these environments. The other types of developers are from scientific organizations or universities, or industry consortiums (or individual companies) that, unlike the Armed Forces, don’t have a military point of view. Instead, they create methods for scientific development, or large scale commercial deployment.

The third parameter of Table 3-2 is the type of vehicle. This is obviously one of the most important characteristics of the communications methods because there are some methods that are generic in terms of the environment of the vehicle and others that are more specific for a certain type, like CompactCL which is specific for UUVs. Therefore, this parameter can be divided in four types: UAVs; USVs; UUVs; UGVs; and UxVs, which includes every other type.

The next parameter in Table 3-2 is accessibility of the IBB. This is important because, for example, there are some IBBs that are only for military forces or NATO countries and are not available for the civilian markets. Others are licensed, such as Berkeley Software Distribution (BSD), which is a permissive free software license, or Lesser General Public License (LGPL), which is also a free software license. Therefore, this is an important characteristic for any developer that wants to choose between different methods. There are the open IBBs, whose specification and/or software are openly available to the public. Some of these have open-source examples, and others don't.

The last parameter in Table 3 2 is importance of IBB. This is an important parameter because there some methods that have more importance than another's an the most important are JAUS, ROS and MAVlink (+++) after these ones the next more important are STANAG4586, MOOS, NIAG SG – 147 (++) and after these ones all the others.

As previously stated, Table 3-2 compares each IBB according to its purpose. The next step is to compare each standard, data model, framework and protocol with the other IBBs of the same type. The next sections present these comparisons.

3.2.3. Comparison of Standards

Table 3-3 specifies characteristics for each of the standards. The parameters are explained in the next paragraphs.

Table 3-3 - Standards Comparison

Standard	Purpose	Language Support	Open-Source Code
STANAG 4586	Whole system	-	No
JAUS	Whole system	C++	Yes
MOOS	Communications	C++	Yes
CompactCL	Communications	C and C++	Yes
CommonCL	Communications	-	Yes
CLARAty	Whole system	-	No
ECOA	Whole system	-	No

As previously referred, standards are classified as being generic architectures, either for the whole system or only for communications. Therefore, the first parameter of Table 3-3 is the purpose. Standards classified as "whole system" are those that characterize and define the whole architecture, and not only the communications layer. On the other hand, MOOS and CompactCL focus more on the communications architecture, being classified only communication issues.

The second parameter of Table 3-3 is the language support. This is an important parameter for any developer, as it should be considered when choosing the appropriate standard. The standards reviewed have either no language support, or support C or C++. STANAG 4586, CommonCL, CLARAty and ECOA do not present any language support, meaning that there is no native support available for any programming language, and the developer can implement it using the programming language he wants, such as C, C++, python, among others.

The final parameter in Table 3-3 is the open-source code. This is a very important parameter for any developer, as it addresses the possibility of having open-source code to work with, buying it from a propriety vendor, or starting the development from scratch. There are standards that do not present open-source code, such as STANAG 4586 (that relies almost exclusively on propriety software), and there are others that have open-source code, such as JAUS, with the OpenJAUS implementation.

3.2.4. Comparison of Data Models

Table 3-4 introduces data model comparisons.

Table 3-4 - Data Models Characteristics

Data Model	Doctrine	Purpose
BML	Military	Command and Control
MAJIIC	Military	Exchange of data
AVCL	Maritime	Exchange of data
NIAGSG-157	Generic	Command and Control

As previously stated, a data model can be defined as abstract way on how data is represented in the communication system. Therefore, each data model is designed for a certain environment.

The first parameter in Table 3-4 is the doctrine in which the data model is based. There are some data models that are specific for military doctrine, such as

BML, which was designed by the U.S. Army. On the other hand, there are others that focus on the maritime environment and additionally, they are not only for the military, but also, for the industry at large too. Finally, there are the generic data models, which were designed for any environment, and they have the possibility to be used in military or civilian applications, although they were developed with a military approach.

The final parameter of Table 3-4 is the purpose of the data model. There are some that specify the doctrine that should be used in order to command and control the UxV, such as BML. On the other hand, there are others that were designed to provide the exchange of data, and not only the command and control, such as MAJIIC and AVCL.

3.2.5. **Comparison of Frameworks**

Table 3-5 introduces the comparison between frameworks.

Table 3-5 - Framework Comparison

Framework	Language Support	Community Support
ROS	C++ and Python	Large
LCM	C, Java and Python	Small

Two frameworks were presented: ROS and LCM. As previously stated, a framework can be defined as a support structure intended to guide the building of something. ROS is an open-source framework, and it can be used in many different programming languages, such as C++ or python. It is also widely used in the research community, having a large support. LCM is a smaller framework when compared to ROS, and it is designed for message passing and data marshaling. It also provides bindings in many languages, such as C, Java and Python. However, it does not have such a large support when compared to ROS.

3.2.6. **Comparison of Protocols**

Table 3-6 introduces the comparison between protocols

Table 3-6 - Protocol Comparison

Protocol	Language Support	Community Support
MAVLink	C++ and Python	Large
IMC	Java and C++	Small

Finally, there are also two protocols reviewed, which are MAVLink and IMC. MAVLink provides lightweight communications, and it is focused on the exchange of messages for MAVs. It can be used in languages such as C++ and python, and it has a large support in the research community. IMC is also a message-oriented protocol, designed to have communication between heterogeneous vehicles. It does not have such a wide support as MAVLink, but it can also be used with different programming languages, such as Java or C++.

In conclusion, there are many IBBs that can be used to fulfil the requirements of the researcher. However, there are some IBBs that are easier to adapt to any project, as they are broader. The JAUS standard is an example of that. JAUS is a standard that can be used in any vehicles, and it has open support services. It also has open software, such as OpenJAUS[130], which is a great tool to get started and to try this standard. All these characteristics make JAUS one of the most used IBBs in the UxS market.

4

RAMP – Our Proposed Reference Model

It would be normal to present our proposal of a Reference Model for Unmanned Vehicles after reviewing existing standards, data models, frameworks and protocols, commonly referred to as Interoperability Building Blocks (IBB), as we have done in the previous chapter. However, although chronologically this model was developed after a lot of experience and insight gained with those IBB, we chose to present it in this chapter, so that we can refer to it when reviewing those IBB. In doing so we hope to achieve one of the main goals of this thesis: to compare the different IBB using a common model.

As explained in chapter 1, we feel that giving a name to our model is important so that it may be referred to in a simple way. The chosen name was *RAMP,* that stems from the initials of "Reference Advanced Model from Portugal". The name reflects our hope that this model can be a launching pad for a faster, more sustainable growth and comprehension in the area of unmanned systems, much like the common OSI model did in the area of computer networks.

There are several views of what the components of a UxS are, and how these components interact. There is always, at least implicitly, a reference model when describing a UxS. While the models may be different there is a large overlap amongst them. Even when describing very specific UxS (such as UAV, UGV, *etc.* for specialized tasks), it is consensual that some elements, such as the concept of platform and control station are shared amongst them (*e.g.*[93]).

In RAMP we divide the various components in a hierarchical taxonomy (Figure 4-1) composed of:

1) Main Blocks (MB)
2) Main Systems (MS)
3) Sub-Systems (SS)

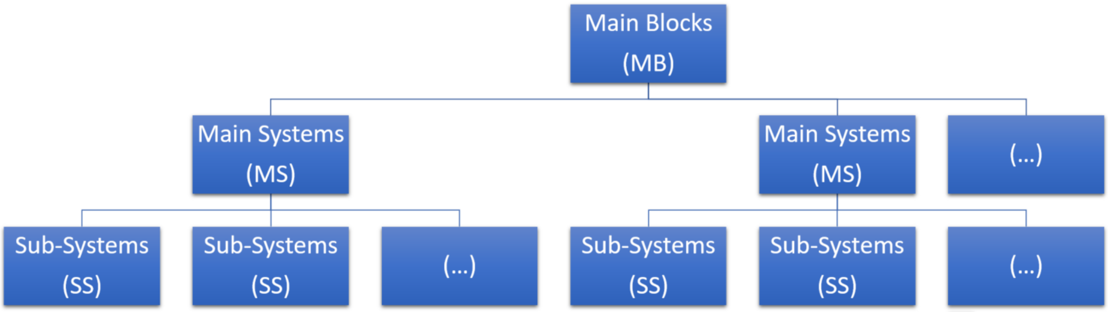

Figure 4-1 - Hierarchical taxonomy

In the RAMP taxonomy there has to be room for new developments, so in all ordered lists, there is always a last item names "others". In some cases, such as when we describe energy sources in MS3.SS1.7, we explicitly name the "others" block and actually make some comments on what it may contain. However, in most cases the "others" item is implicitly the last one and is not explicitly mentioned.

In RAMP there are three Main Blocks (MB) (Figure 4-2):

MB1 - **Vehicle.** This includes everything that is normally onboard the vehicle, *i.e.* all its subsystems, such as payload, navigation subsystem, sensors, communication subsystem, power and propulsion. In some cases, such as when the vehicle is under direct remote control, certain Vehicle sub-systems, such as navigation, may physically be on the GroundSegment.

MB2 – **Datalink**. This includes all that serves a communication path. It establishes a link between the control station and the vehicle, through their both communication subsystems, and may also establish communications with other vehicles or multiple ground stations.

MB3 – **GroundSegment**. The Ground Segment (written on purpose as a single word) includes all the physical components that are outside the Vehicle. These are usually on the ground, but may very well be aboard a ship, a plane, a spaceship, or anywhere else. It will usually be composed of launch and recovery equipment, support equipment, a control station and a communication subsystem.

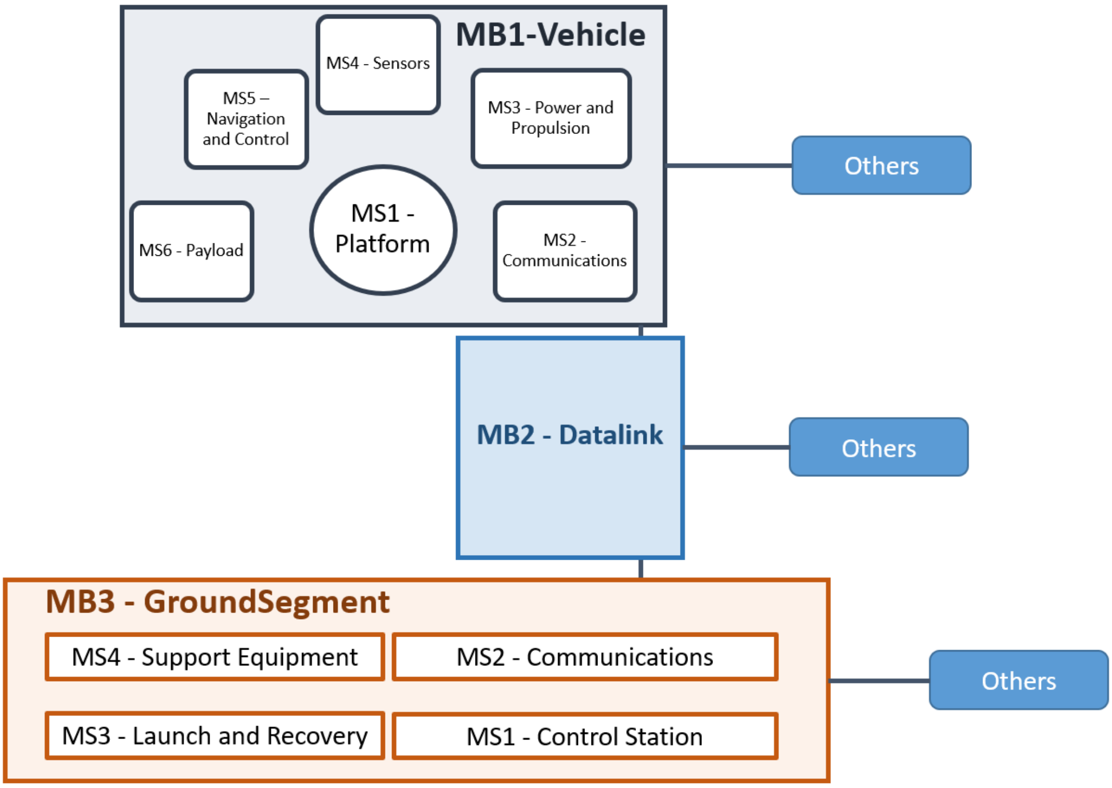

Figure 4-2 - RAMP Main Blocks

RAMP Main Blocks for an unmanned system, with their functional subsystems.

The second level in RAMP is the Main System (MS) level. This is a functional description of the elements that compose the Main Blocks, and are associated with each MB, where they are normally physically located.

The third level in RAMP is the Sub-System (SS) level. This is also a functional description of elements, that in this case are components of the Main-Systems.

4.1. Vehicle Components (Main Systems - MB1.MSx)

The vehicle itself has several Main Systems. All vehicles that we can think of all the 6 main systems described in the RAMP taxonomy, however simple of sophisticated they may be. We shall now describe these 6 main systems, and their sub-systems and components.

4.1.1. **MB1.MS1 - Platform**

The platform (Figure 4-3) is the physical skeleton of the UxS and is responsible for accommodating all the necessary components required for the system to work and do is functions. On a UAV, the platform is the airframe, on a USV it is the hull and superstructures, and on a UGV it is the vehicle itself. The platform is thus very specific to the environment where the UxV will operate, and to the tasks it will perform.

When designing the platform, it's necessary to have many considerations such as the materials used, that may have some particular requirements like lightness, robustness, flexibility, among others. The shape of the platform has to be adequate for the desired purpose, especially when aerodynamics and hydrodynamics are factors must be considered, and it has to house all other vehicle Sub-Systems. We have already discussed the various types of platform in chapter 2 and will not discuss platforms further.

Figure 4-3 - Example of a MB1.MS1 – Platform for a UGV

Photographed at the Portuguese Naval Academy's Robotics Lab.

4.1.2. **MB1.MS2 - Communications**

Any UxV, with whatever degree of autonomy (from purely remotely piloted to almost completely autonomous that only receives general objectives) must communicate with the outside world[208]. The entities with which it must communicate (which we shall call interlocutor) include the other elements of the UxS, namely the ground segment, other vehicles that belong to its system and are thus coupled in some way to the UxV, other vehicles that might be friendly, hostile, or neutral, and other systems, manned or unmanned.

For our reference model, the most important interlocutor is the ground segment, from which it receives its orders and to which it reports the results of the mission. This may be done "offline", i.e. before the mission starts and after it ends, as is common in most UUV, or "online", as is more common on UAVs, where orders are passed on during the mission and results are immediately relayed to the ground segment. In any case, some level of "tasking", from very abstract objectives to specific orders to control surfaces is always given to a vehicle, and some sort of reporting is always sent back, either during the mission or when it ends. The different levels of communication with the ground segment will be discussed in MB3.

When the UxS comprises more than one vehicle, namely when swarms of vehicles are used, communications with other vehicles becomes an important issue to assure the common mission is accomplished[209]. Still within the UxS, the ground segment might have more than one ground control station that is interlocutor for the UxV.

It may also be necessary to communicate with interlocutors outside the UxS. This can be done to answer to traffic control entities, other vehicles (manned or unmanned), etc.

The communication main system must ensure that all necessary interlocutors can be addressed. If no "online" or real-time communication is required, the MS2 may be a simple electronic interface, such as Recommended Standard (RS) 232 or Universal Serial Bus (USB) port, or even just a memory port (such as Flash-Memory or SD card port). The tasking and reporting can also be done without physical contact using optical (usually laser) systems, however in the vast majority of UxS the communication system comprises a radio and an antenna, that as we shall see in the next Main Block, adhere to a given communication standard[210].

The communication main system comprises all communication systems with external entities, and in many cases, is composed of separate systems for platform control and for payload control, normally using different electromagnetic spectra bands.

Naturally, the main communication system (MB1.MS2) of the vehicle is tightly coupled with the MB2 (datalink and communications) and the main communication system of the ground segment (MB3.MS2), and we will discuss it further when addressing them.

4.1.3. **MB1.MS3 - Power and Propulsion**

UxS power and propulsion system can be functionally divided in 6 Sub-Systems that do not have to be present in all Power and Propulsion Systems: **SS1 - Energy Source**, **SS2 - Energy Transformer**, **SS3 - Powerplant**, **SS4 – Mechanical Coupling**, **SS5 - Propulsion Effector**, and **SS6 - Control Effector** (Figure 4-4).

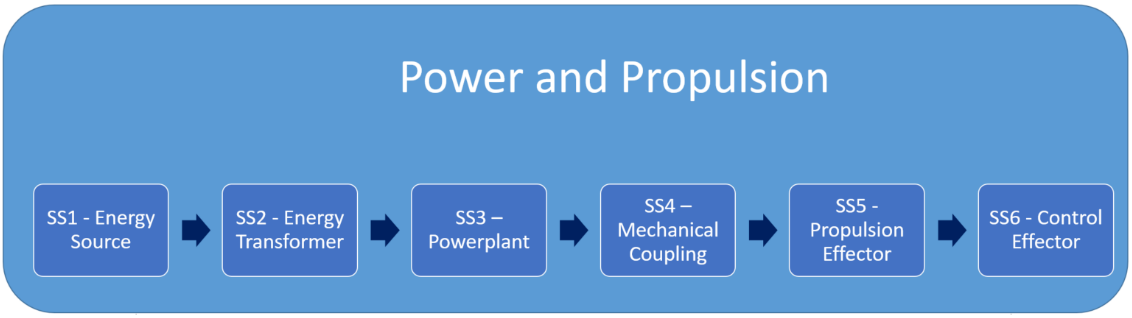

Figure 4-4 - Conceptual view of an UxS power and propulsion system

Source: [211]

Before delving into the specifics of the level 3 sub-systems, we may consider some broad types of Power and Propulsion systems that can be categorized in various ways.

Regarding their dependence on internal or external power sources, we can group them in:

- **Internal energy systems**, that rely mainly on fuel available on the vehicle before the mission starts. This includes internal combustion engine systems, rocket systems, electric systems relying on batteries or fuel cells, *etc*.

- **Energy Harvesting Systems**, that try to draw energy from the environment, such as aerial gliders, sailing vessels, solar powered systems, wind generator systems, or temperature gradient underwater gliders.

Regarding the type of propulsion system, and with great variations from UAV, USV, UGV, and UUV, we can group them in:

- Propeller Aerial Systems;
- Jet Aerial Systems;
- Rocket Aerial Systems;
- Propeller maritime systems (both for USV and UUV);
- Wheeled Ground Systems;
- Tracked Ground Systems;
- Biomimetic Systems, that depending on the medium can be:
    - Flapping wing Aerial Systems;
    - Undulating Underwater or Surface Systems;
    - Multi-legged Ground Systems;
    - Pendular Systems.
- Others

4.1.3.1. **MB1.MS3.SS1 - Energy Source**

The energy source can vary between gasoline, diesel fuel, lithium-hydride, liquid hydrogen, solar energy, wave energy, among other types of fuel, providing system's energy. We can divide the energy sources into the following broad classes (see Figure 4-5)

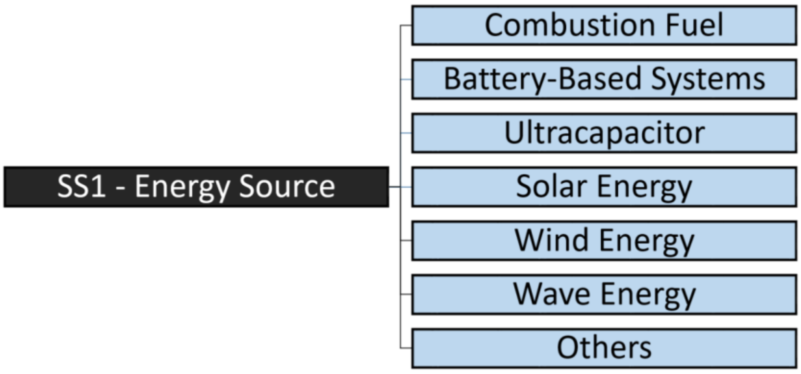

Figure 4-5 -Energy Source classes.

4.1.3.1.1. MB1.MS3.SS1.1 - Combustion Fuel

This is the most common energy source for large systems. This fuel is normally in liquid form and stored in tanks. Common fuels are standard gasoline, diesel fuel, kerosene, naphtha, JP10, otto fuel, ethanol, liquid hydrogen, etc[211].

4.1.3.1.2. MB1.MS3.SS1.2 - Battery-Based Systems

Battery-Based Systems (Figure 4-6) are based on the interaction between anode and cathode through a conductive electrolyte. This interaction generates an electron flow through a connected low, providing power that generates motion. Regarding UxS, rechargeable batteries are the most conventional. It grants the advantages of being silent, lightweight, efficient, no waste, self-contained, non-vibrant, rechargeable and reliable. It has the down side of having limited endurance, inefficient recharging process, internal resistance heating, performance sensible to surrounding temperature and it carries corrosive chemicals[212].

Figure 4-6 - Example of a MB1.MS3.SS1.2 –Battery SW1870.

Photographed at the Portuguese Naval Academy’s Robotics Lab

4.1.3.1.3. MB1.MS3.SS1.3 - Ultracapacitor

The Ultracapacitor is a capacitor which can store huge amounts of energy. A capacitor is an electrical component capable of storing energy in an electrical field. It does this by having electrical conductors separated by an isolator (dielectric material). When a voltage is applied on the conductors, electrostatic charges accumulate on the conductors. One of the advantages of ultracapacitors over batteries is that they are able to provide high power charge quickly[213].

4.1.3.1.4. MB1.MS3.SS1.4 - Solar Energy

Solar Energy is a very popular source for long endurance systems, mainly because a photovoltaic solar panel produces electrical energy that is easily stored in batteries[214]. Amongst the examples we have a vehicle developed at *École Navale* (Brest, France) (Figure 4-7).

Figure 4-7 - Example of a MB1.MS3.SS1.4 –Solar power vehicle.

Developed at the French Naval Academy in 2008 and used for various tests

4.1.3.1.5. MB1.MS3.SS1.5 - Wind Energy

Wind is widely available for USV. Some are classical sailing boats, but rigid sails, vertical rotors, kites, and other wind harvesting systems have been used. For UGV wind is not usually used, but there are exceptions ("char-a-voiles"). Aerial gliders are also common, mostly for systems that are deployed for short periods (launched from aircraft or from the ground), but some long-endurance gliders have also been proposed[215].

4.1.3.1.6. MB1.MS3.SS1.6 - Wave energy

The most popular USV that uses wave energy is the Wave Glider by Liquid Robotics (see Figure 2-23), but other implementations of the concept exist[216].

4.1.3.1.7. MB1.MS3.SS1.7 - Others

New energy sources appear every day, such as using animals, bioconverters, etc[217]. We thus allow a generic class of "other energy sources" to complete our taxonomy.

4.1.3.2. MB1.MS2.SS2 - Energy Transformer

This subsystem only occurs in certain types of Power and Propulsion Systems. One such case is when fuel cells are used: the energy source is hydrogen, stored in liquid form or in a metal hydrate, but it has to go to an Energy Transformer (in this case a fuel cell) to be converted into electricity that can be used by the powerplant. When using energy harvesting systems, it is the Energy Transformer (solar panel, sail, wind generator, *etc.*) that is perceived as Source, but the true source is the environment. Even classical fuel systems, such as naphtha or even diesel oil, may need energy transformer systems, such as a pre-heating unit or fuel centrifuge. The energy transformers can be classified as show in Figure 4-8.

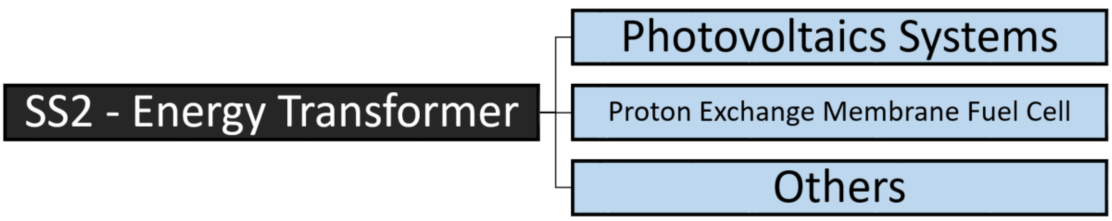

Figure 4-8 - Energy Transformers classification.

4.1.3.2.1. MB1.MS3.SS2.1 - Photovoltaic Systems

Photovoltaic Systems (Figure 4-9) use the photoelectric effect to obtain electric current. Electrons are emitted with the absorption of electromagnetic radiation from the sun, using solar cells. Solar panels are silent, they do not induce vibrations, and are very reliable and low-maintenance. However, they tend to be expensive, their efficiency is low, and require sun (that may not be available) and backup batteries that may not be very durable[218].

Figure 4-9 - Example of a MB1.MS3.SS2.1 –Photovoltaic systems.

Photographed at the Portuguese Naval Academy's Robotics Lab

4.1.3.2.2. MB1.MS3.SS2.2 - Proton Exchange Membrane Fuel Cell

Proton Exchange Membrane Fuel Cell involve an electrochemical oxidation process to generate electric current. This process is based on the ionization of hydrogen, which separates into protons and electrons. Only the protons are allowed through an electrolyte membrane, from the anode to the cathode. After passing through the membrane, the protons combine with oxygen in a process which requires electrons. Thus, the electrons meet this requirement by passing from the anode to cathode through a load, generating electric current. It is a promising technology and it's more efficient than combustion[219]. It is also a quiet process, it has no vibrating parts, zero- emission, higher energy density than batteries. It also has disadvantages: it is expensive, it has pressurized components, it is complex when compared to batteries, sensitive to water and humidity and it is a still developing technology.

4.1.3.2.3. MB1.MS3.SS2.3 - Others

Energy Transformer systems can be quite varied, so allow a generic class of "other energy transformers" to complete our taxonomy.

4.1.3.3. MB1.MS3.SS3 - Powerplant

The powerplant also acts as a transducer, and the final product is motion. It collects the transformed energy to develop motion to create propulsion, as in an expanding combustion chamber. There are many types of powerplants than can be used in UxS, but the main ones are listed below and shown in Figure 4-10.

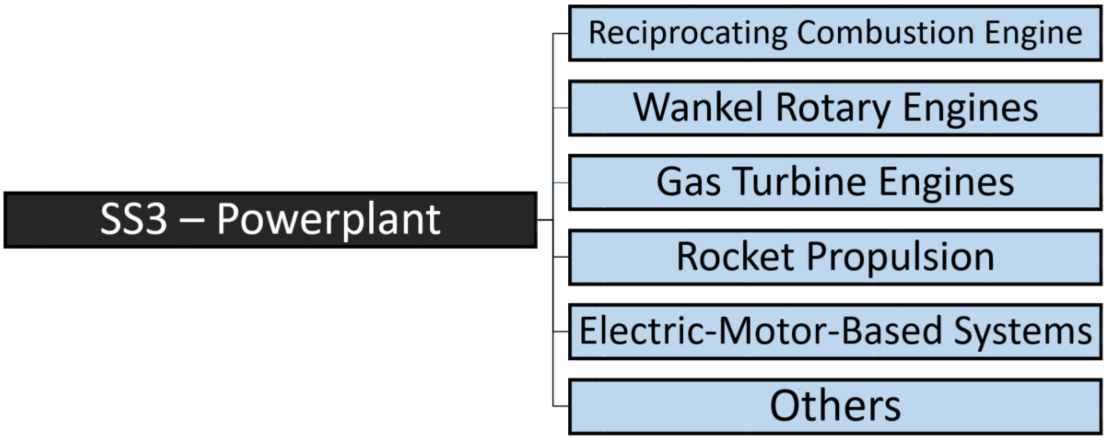

Figure 4-10 - Types of Powerplants.

4.1.3.3.1. MB1.MS3.SS3.1 - Reciprocating Combustion Engines

The Reciprocating Combustion Engines use pistons contained in cylinders to generate movement, through the combustion of fuel. The intake, compression, explosion and exhaust phases generate force which is distributed as motion. This technology has the advantages of being: widely understood technology, allowing energy efficient diesel engines, potentially lightweight and potentially small sized. On the other hand, they can be noisy and generate vibration, and may require sealing, lubricating and cooling systems.

4.1.3.3.2. MB1.MS3.SS3.2 - Wankel Rotary Engines

The Wankel Rotary Engines are, in concept, like the previous engines but they differ on how the combustion is generated. The previous engine's combustion was generated by an up and down movement of the pistons inside a combustion chamber, while in this case the combustion is generated by rotary movements inside a combustion chamber. These engines have the advantage of having: higher power output for similar displacement, thus smaller size; iron rotor in aluminum housing that reduces likelihood of engine seizure; lighter weight than legacy or compression-ignition engines; less noise and vibration than reciprocating engines; and reliability close to that of a turbine. On the other hand, their liquid-cooled engine may turn them heavier and more complex; their fuel consumption is higher than diesel engines; and they have trouble in meeting emission standards[220].

4.1.3.3.3. MB1.MS3.SS3.3 - Gas Turbine Engines

Gas turbine engines can be of various types like jet turbine, turbofan and turbo propeller engines. They are based on a dynamic internal combustion process involving the passage of air and fuel at different velocities and pressures through many rotary small blades. This process originates high power thrust which allows a vehicle to achieve high speed. It has the advantages of having high power density, great thrust capability, achieving supersonic velocities, efficient at small loads, almost insensitive to fuel quality, no need for lubricating fluid. On the other hand, they are expensive, loud, complex, achieving high rotation speeds and high internal temperature.

4.1.3.3.4. MB1.MS3.SS3.4 - Rocket Propulsion

Rocket propulsion is obtained by chemical reaction that results in tremendous pressures forcing high velocity particles through a nozzle, which provokes great impulse, generating motion. It allows for high power density and a self-contained energy source for low-oxygen environments. However, it's inefficient at low speed, it has a high fuel consumption, and a complex and expensive control and guidance system[221].

4.1.3.3.5. MB1.MS3.SS3.5 - Electric Motor-Based Systems

Electric motor-based systems (Figure 4-11) originate movement through the use of electromagnetic interaction between stator and rotor. This type of engine is easily found in all types of UxS, ranging from the larger to the smaller ones. They have the advantage of having low maintenance, high reliability, robustness, low risk of overheating, and high torque at low speeds. However, they are vulnerable to electromagnetic interference, require large currents and are sensitive to water and other conductive liquids[222]. Besides being used for main propulsion, electric motors are used a lot in most UxS to power different types of actuators, such as fins, rudders, ailerons, etc. There has been a great deal of development with electric motors in recent years, due to the widespread use of brushless motors and advanced solid-state control systems.

Figure 4-11 - Example of a MB1.MS3.SS3.5 –Electric Motor AMPFlow.

Photographed at the Portuguese Naval Academy's Robotics Lab

4.1.3.3.6. MB1.MS3.SS3.6 – Others

Many different powerplants exist, such as Sterling engines, steam engines, etc. We thus allow a generic class of "other powerplants" to complete our taxonomy.

4.1.3.4. MB1.MS3.SS4 – Mechanical Coupling

This subsystem only occurs in certain types of Power and Propulsion Systems. Classical examples of these systems are gearboxes in many types of UxS, large shafts in USV, magnetic couplers in UUV, *etc.* These can be classified as shown in Figure 4-12.

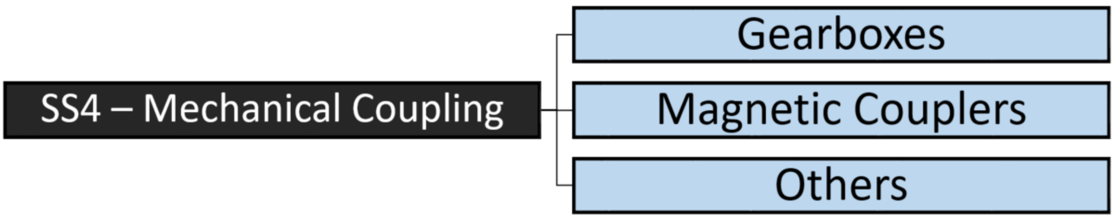

Figure 4-12 - Mechanical Coupling classification.

4.1.3.5. **MB1.MS3.SS5 - Propulsion effector (MB1.MS3.SS5)**

Propulsion effectors are the devices that produce the motion of the vehicle. Again, the type of effectors available depends a lot on the type of environment, as shown in Figure 4-13.

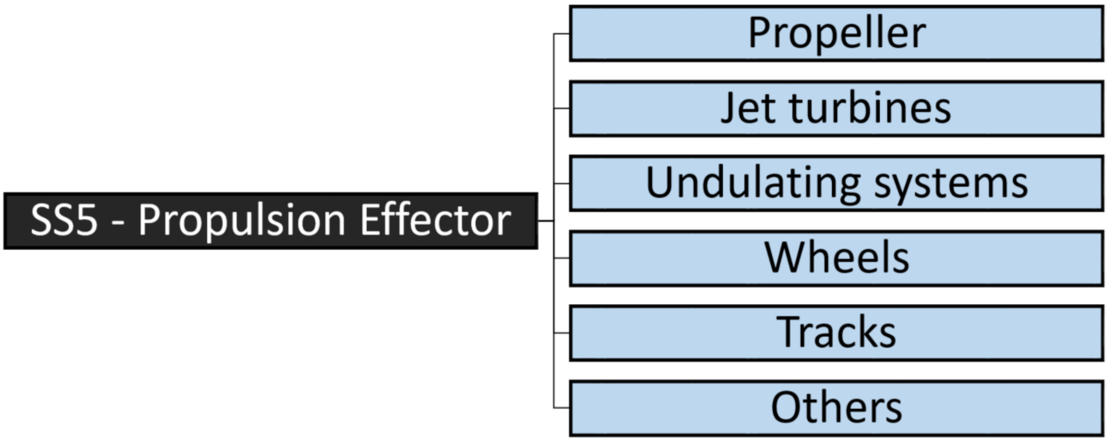

Figure 4-13 - Types of Propulsion Effector.

4.1.3.5.1. MB1.MS3.SS5.1 Propeller

Propeller-based systems push the external fluid in the desired direction, transforming a rotary motion of a shaft into a forward motion of the vehicle. These systems are widely used and have the advantage of having: low engineering overhead; simpler control than a turbine; low cost compared to other thrust-based systems; and faster reactions to control alterations. On the other hand, they are a danger to personnel or objects in the vicinity, their efficiency varies with rotation speeds, and are sensitive to and produce vibrations[211].

4.1.3.5.2. MB1.MS3.SS5.2 Jet turbines

Jet turbines are seldom used (with notable exceptions such as the Global Hawk) because they operate better at high speed and are less efficient than propellers (even if the power itself comes from a turbine)[223].

4.1.3.5.3. MB1.MS3.SS5.3 Undulating systems

Undulating systems, or more generally biomimetic systems, are becoming more popular[224] and mimic propulsion systems used by animals. In the case of undulation propulsion, a (usually soft) surface moves back and forth producing forward thrust. These systems have been used mainly in UUV, but they can also be used in UAV that flap their wings like birds[225], or UGV that move like snakes. Other biomimetic systems include multiple legged systems, or squid-like systems.

4.1.3.5.4. MB1.MS3.SS5.4 Wheels and Tracks

For UGVs, wheels are probably the most common effectors, sometimes in ingenious configurations[226], but tracks have the advantage of having a lower load and being able to move in very irregular surfaces.

4.1.3.5.5. MB1.MS3.SS5.5 Others

Other effectors include pendular systems, rotating vehicles, etc.

4.1.3.6. MB1.MS3.SS6 - Control Effector

Besides the propulsion itself, most vehicles have what we called "control effectors" that in some way change the motion of the vehicle. Simple examples are steering wheels for UGVs, rudders for USVs, or ailerons for UAVs. We have included them in the Power and Propulsion Main System because they affect the motion of the vehicle. The main control effectors are presented in Figure 4-14.

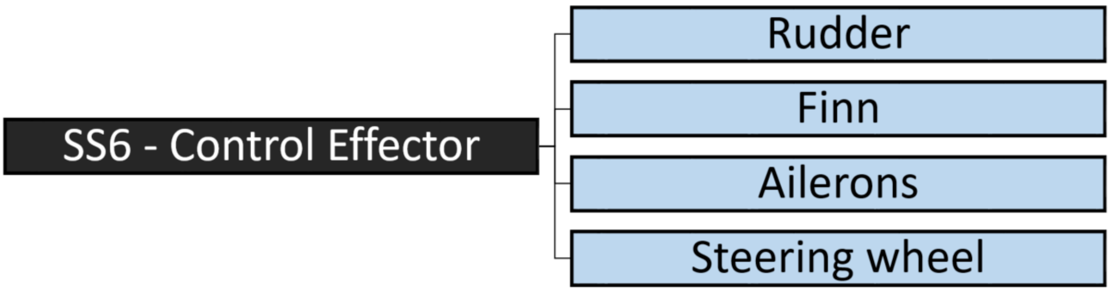

Figure 4-14 - Main Control Effectors.

4.1.4. MB1.MS4 - Sensors

A sensor is an equipment that receives a physical stimulus and responds with a signal (typically electrical). That signal can then be used to estimate a given property or characteristic. There are various ways to classify sensors. They can be classified according to the property or characteristic we want *to know*, or according to what *physical stimulus* they measure. As an example, according to what we want to know, we may use an altimeter to estimate how high we are flying. According to the physical stimulus, we may have an air pressure sensor (since the altitude is inversely proportional to air pressure), or a system that measures the flight time a radio wave takes to go from the UAV to the ground and bounce back to the UAV. The same final information (altitude) may be obtained from sensors that use vastly different physical stimulus. On the other hand, those same sensors (air pressure and radio flight-time sensors) may be used to obtain completely different information, since an air pressure sensor can be used as a speedometer (in a pitot tube) and a radio flight-time sensor can be used for object detection (as a RADAR). When we classify the sensors according to what we want to know we are using a *functional classification*; when we classify them according to the stimulus they measure (or how they measure it), we are using a *physical classification*. Furthermore, we can classify them according to the final objective of the information, for it may be for platform control (and thus it is a *platform sensor*), or it may be the objective of the mission (and thus it is a *payload sensor*). We may call this a *utility classification*.

There has been a lot of work on sensor taxonomy in various areas[227],[228]. There are some IEEE work groups in this area, from where standards such as the SensorML[229] extension of XML have emerged. Standards like this, or the Open Geospatial Consortium OGC's Sensor Web Enablement (SWE)[230], have a huge influence on how the industry categorizes sensors. These different taxonomies may be very encompassing or very well suited to specific domains. Unfortunalty, we did not find a taxonomy that is specific enough, and at the same time sufficiently broad, to be of use in RAMP. Therefore, we developed our own taxonomy.

For the RAMP taxonomy, we use mainly a functional classification of the sensors. However, a utility classification is implicit since the same sensors may appear as payloads, where they have different implications. Furthermore, a physical classification may be necessary, for example due to electromagnetic compatibility or stealth issues (measuring altitude using RADAR might be unacceptable for a military surveillance UAV).

We shall now list the most important sensor subsystems (MB1.MS3.SSx)(Figure 4-15), named after the information we want from it (i.e., using a *functional classification*)

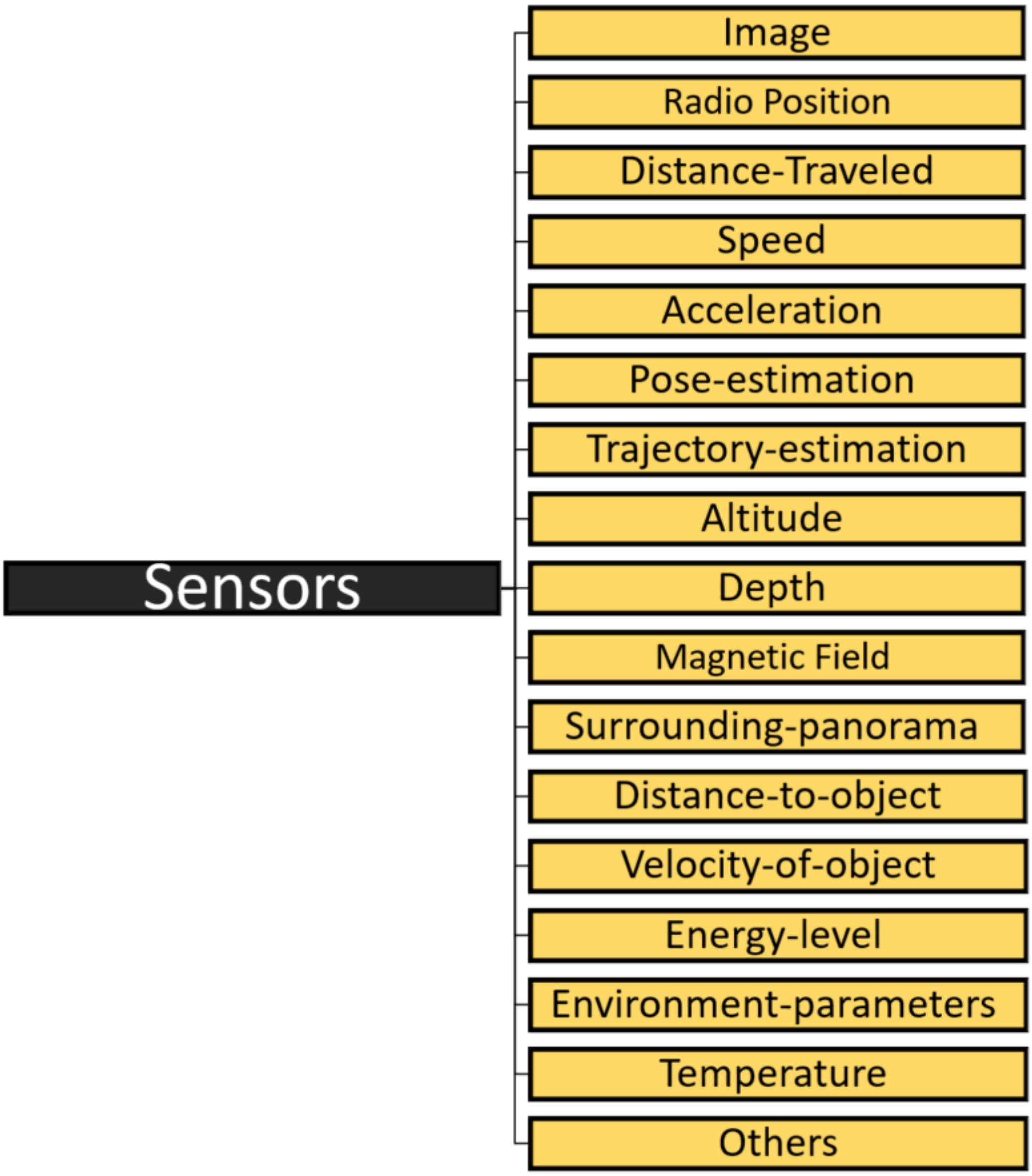

Figure 4-15 - Sensors

4.1.4.1. MB1.MS4.SS1 - Image (visual, infra-red, spectral)

Image sensors, commonly known as cameras (see Figure 4-156), are probably the most ubiquos sensors in UxV, because all humans like to see what is happening in the UxVs environment. Even in UUV, that operate in environments with very poor visibility, cameras are common. Cameras vary widely, depending on: the electromagnetic band they operate in (visual, infa-red, near-infrared, multispectral, etc); the spatial resolution of the image, normally measured in pixels; color depth; distortion (highly depend on the lens); sensitivity; etc[231].

Figure 4-16 - Example of a MB1.MS4.SS1 –Camera National Instruments NI 1722.

Photographed at the Portuguese Naval Academy's Robotics Lab

Image sensors are closely related to surrounding-panorama sensors, but those will usually require active sensing of the environment, while image sensors are mainly passive (although they might use in some cases flashes)

4.1.4.2. **MB1.MS4.SS2 - Radio position (GPS, LORAN-C, Radio-Beacons)**

Most UxV use some sort of external radio reference to estimate their position and enable navigation. The navigation methods themselves will be discussed later (in MB1.MS5 – Navigation and Control), but many of them rely on some sort of sensor that receives radio signals. The sensors themselves are usually just antennas, positioned so that they don't get interference from on-board systems[232]. The most common radio position sensors are GPS antennas (Figure 4-17), which are quite small and simple. Radio goniometers are generally larger but are necessary to determine the direction of radio-beacons.

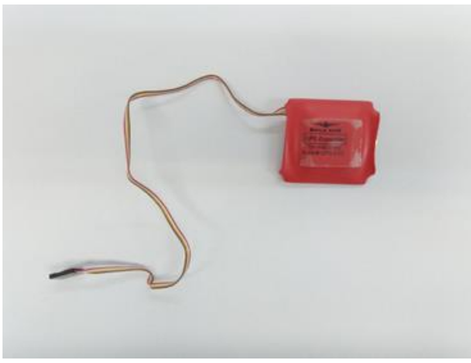

Figure 4-17 - Example of a MB1.MS4.SS2 –GPS Eagle Tree.

Photographed at the Portuguese Naval Academy's Robotics Lab

4.1.4.3. MB1.MS4.SS3 - Distance-Travelled (1-dimensional odometers)

Distance-Traveled sensors, or odometers, usually count the rotations of wheels (either wheels of UGV, water threads in USV and UUV, or propeller rotations in UAV), but the same type of sensor can be obtained by integrating speed sensors (pitot tubes, acoustic odometers) or double-integrating acceleration sensors (MEMS accelerometers and gyroscopes)[233].

4.1.4.4. MB1.MS4.SS4 - Speed (1-dimentional speedometers, variometers, etc)

Speed sensors, or speedometers, may measure velocity using different principles. Pitot tubes measure air pressure, acoustic speedometers measure doppler distortions, RADAR sensors also measure doppler distortions, etc. In aviation, vertical speed sensors are called variometers, and usually use variations in air pressure to determine vertical velocity[234].

4.1.4.5. MB1.MS4.SS5 - Acceleration (1-dimentional, 3-dimentional, or 6-dimentional accelerometers and gyroscopes)

Acceleration sensors, or accelerometers are mainly inertial sensors that measure forces induced in masses, but optical accelerometers, and even quantum-sensors (that are getting ever more popular) may be used.

4.1.4.6. MB1.MS4.SS6 - Pose-estimation (inertial sensors)

Pose-estimation sensors are usually acceleration sensors that measure gravity to obtain a vertical reference, and possibly the earth's magnetic field to obtain the horizontal direction. However, pose-estimation may be done using infra-red or visual sensors to detect the horizon line (and thus obtain a vertical reference). Pose estimation can also be obtained by integrating accelerometers[235].

4.1.4.7. MB1.MS4.SS7 - Trajectory-estimation (IMU, integration sensors, DVL)

While odometers will usually provide information about the distance traveled along a single, or multiple axis, a trajectory-estimation sensor will provide a path along a 2-dimentional or 3-dimentional space. This can be done by integrating inertial, velocity sensing, or distance travelled devices, or by terrain following techniques using Doppler Velocity Log (DVL)[236] or Simultaneous Localization and Mapping (SLAM) systems.

4.1.4.8. **MB1.MS4.SS8 - Altitude (altimeters, from ground, sea-level, or sea-bottom)**

An altimeter (Figure 4-18), as the name indicates, measures the altitude above a fixed level. Barometric altimeters are the most common[237]. These devices measure the absolute air pressure, and assuming a given pressure at sea-level and air density, estimate the altitude. Because these altimeters rely on air pressure information, and the correlation with altitude varies with external factors like temperature, they need to be calibrated according to weather conditions in order to give reliable information. Radio altimeters measure the flight time of radio waves to bounce off the surface, and thus measure the altitude relative to the ground at that point, but they are very accurate. Ultra-sound or sound altimeters use the same principle, and thus also measure altitude relative to the ground at that point. In maritime vehicles, SONARs are commonly used as altimeters to know the distance to the sea-bottom.

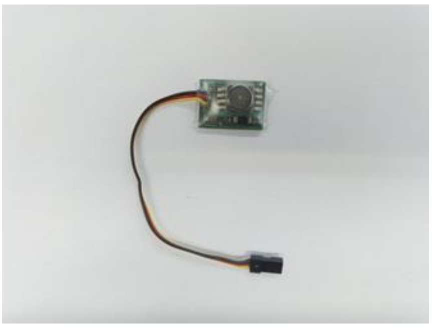

Figure 4-18 – Example of a MB1.MS4.SS8 –Barometric altimeter 1A DMD.

Photographed at the Portuguese Naval Academy's Robotics Lab

4.1.4.9. **MB1.MS4.SS9- Depth (depth meters, from the water surface)**

In this taxonomy, "depth" refers to the depth of the vehicle relative to the sea-surface. What is commonly known as a Depth Sounder (a device that uses acoustic signals to measure the depth under the sensor by detecting the sea floor or underwater objects) is in fact an altimeter in this taxonomy. This is necessary to assure coherence across a vast range of vehicles: depth is the distance *up*, away from the centre of the earth, from the vehicle to the sea surface (or ground surface for a hypothetical tunnelling ground vehicle); altitude is the distance *down*, from the vehicle to sea-level, ground-level, or sea-floor. Thus, if we want to know the "water depth" under a maritime vehicle, we need an altimeter (referenced to the sea-floor).

Depth sensors are usually pressure sensors that measure the weight of the water column above the vehicle. Since water is far denser than air, depth sensors tend to be far more accurate than barometric altimeters, because pressure variations at the surface and density variations in the water columns will induce only minor errors. Depth sensor can, however, be SONARs pointed upwards to the surface of the water.

4.1.4.10. **MB1.MS4.SS10 – Magnetic Field (compasses)**

Almost all UxV have a magnetic compass, that is particular type of magnetic field sensor, or magnetometer. A magnetic compass gives the direction of the UxV relative to the earth's magnetic field (what is known as magnetic bearing). Since in most areas of the globe the magnetic North is quite close to the true North, the magnetic bearing is a quite good estimate of the true bearing and is used instead of it. A traditional magnetic compass is composed of a magnetised body that moves freely on a horizontal surface, but this does not allow an easy interface to an electronic control system. Most magnetic compasses used in UxV are FluxGate sensors mounted on a gimbal (to provide a vertical reference). A FluxGate sensor is basically a transformer that saturates its magnetic core, and the intensity of the magnetic field of the earth is measured by its constructive or destructive interference with the induced field, altering the saturation point of the core. They can be built at a micro-scale, and thus provide a very small and lightweight sensor with no moving parts and easy electrical readings[238].

The main problem with magnetic sensors aboard UxV is the interference of all the electrical equipment and UxV body parts with earths field. Thus, the magnetic sensors themselves are usually positioned as far away as possible from all other devices, such as the tail of a UAV or the mast-top of a USV.

Other types of magnetometers, such Hall-Effect or Superconducting Quantum Interference Device (SQUID) are also used either for vehicle health-management, proximity sensing, or as payload to measure fields produced by other objects (such as mine or submarine hunting)[239].

4.1.4.11. **MB1.MS4.SS11 -Surrounding-panorama (RADARs, LIDARs, SONARs)**

We classified as surrounding-panorama sensors all those that provide some type of map of the surrounding area. A RADAR image or side-scan SONAR image are typical examples.

A RADAR is a system that uses radio waves to determine the range, angle or velocity of contacts in the surround environment. The term was coined in the second world war and referend to "RAdio Detection And Ranging". Most RADAR systems measure the flight-time radio waves take to travel from the antenna to an object and bounce back. If the antenna rotates, an image of the distance to the first object in all directions can be obtained. There are however many variants to this basic principle some of which will be reviewed later because the information they give is not a panorama. A variant of the basic RADAR that is becoming very common is the SAR[240]. These RADARs use the motion of the RADAR antenna relative to a target to provide finer spatial resolution of images that can be either two or three-dimensional representations of the object. Other variants include Multi-beam RADARs, Phased Array RADARs, Bi-static, multi-static, and passive RADARs[241].

Light Detection and Ranging (LIDAR) is an active form of sensing, very similar to RADAR, but that uses light instead of radio-waves. The range is usually much smaller than RADAR, but they are more accurate and generally smaller[242].

SONAR is another equipment similar to RADAR, that uses sound instead of radio-waves. SONARs are used mainly in water, and as with RADAR, many variants of SONAR exist, such as multi-beam, side-scan, SAS (Synthetic Aperture SONAR), passive SONARs, etc. Since light and radio-waves are severely attenuated in water, SONARs are extremely important in all underwater applications[243].

4.1.4.12. MB1.MS4.SS12 -Distance-to-object (directional RADARs, ultra-sound distance sensors, stereoscopic visual sensors)

As the name implies, distance-to-object sensors estimate how far an object is from the vehicle. In some cases, for very close distances, there may be a physical contact (a "sensing rod") between the two. In most cases, this measurement if made at a larger distance using capacitive, inductive, magnetic, optical, sonic, ultrasonic, or radio sensors[244].

Although the physical phenomena used for sensing is very similar to most surrounding-panorama sensors, the way they are used is quite difference. Directional RADARs (commonly known as "attack RADARs" in military jargon) measure the distance to a target using a single "ping". Stereoscopic visual sensors use the difference between images captures at different points to estimate the distance to an object that is contained in both of them[245].

A particular sub-class of distance-to-object sensors are the proximity sensors. While measuring a single distance, these sensors tend to be used only at very short ranges, to avoid collisions, and while some may provide accurate estimates, most will just have a threshold to give a proximity alarm.

4.1.4.13. MB1.MS4.SS13 -Velocity-of-object (CW RADARs, Doppler sensors)

These sensors measure the velocity of an object relative to the unmanned vehicle. This is usually used to track other objects but may be used for the vehicle's navigation when pointed at fixed objects such as the sea-floor, ground, or conspicuous landmarks.

Continuous wave RADARs (CW)[246], for example, measure doppler shifts in the received waves, and thus relative speed of the vehicle and the "target".

Doppler Navigation[247] is the term used for navigation systems, that use either RADAR or SONAR to estimate ground speed. DVL devices are particularly popular for underwater vehicles moving close to the bottom, using a multibeam SONAR.

4.1.4.14. **MB1.MS4.SS14- Energy-level (battery level, tank level)**

For the vehicle's own monitoring, almost all UxS have some type of energy level sensor to estimate how long it can operate and in what conditions. The energy-level sensor can be a simple voltmeter for a battery-operated system, but most modern UxV have complex energy-management units to obtain more rigorous estimates, detect problems, and perform "health-management" services. For UxV that use liquid fuels, tank level sensors are common, using multiple point sensors, ultra-sound sensors, of mechanical level sensors.

4.1.4.15. **MB1.MS4.SS15 - Environment-parameters (anemometers, radiation and chemical sensors)**

Environment sensors are usually used as payload, but in some cases they are also important for the vehicle itself, such as anemometers (see Figure 4-19) for sailing vessels, or chemical sensors to ensure the vehicles safety.

Figure 4-19 - Example of a MB1.MS4.SS14 –Anemometer WindMate WM-200.

Photographed at the Portuguese Naval Academy's Robotics Lab

Chemical sensors measure and detect chemical substances. Traditionally, chemical sensors requited some sort of chemical reaction with the substance being measured and presented the results off-line because they could not be converted into electrical signals in a continuous basis. However, recent developments have led to a large number of specialized sensors that provide real-time measurements of concentrations (or at least existence or not) of chemical substances in the form of electrical (and usually digital) signals. In most cases the chemical agents being measured must be in both in liquid or gas phase and in direct contact with the sensor. Other chemical sensors can also use light or radio signals to detect chemical agents at a distance using spectral analysis.

Radiation sensors detect electromagnetic waves emitted by a foreign body. One of the most common sensor is the Scintillating detector that converts nuclear radiation into light. Other common sensors include Ionization detectors which detect the ions created by radiation. The Geiger-Müller Counter is an example of an ionization detector since it detects the electrons that result from the ionization process caused by radiation (alpha, beta and gamma radiation) to measure radiation levels[248].

4.1.4.16. **MB1.MS4.SS16 - Temperature (contact sensors, IR sensors)**

Temperature sensors (Figure 4-20) are common and essential sensors for modern electronic devices, so as to prevent overheating and damage. They are also crucial, for the same reason, for most propulsion systems. Temperature sensors can also be used as payload, both for monitoring the environment and for obtaining temperature maps of targets. The most common types of temperature sensors are thermocouples[249], thermo-resistors, and thermistors[250],[251]. Temperature sensors may also sense at a distance using Infra-Red radiation (IR sensors). These can be point sensors, or thermal cameras. Besides being used to measure temperatures of objects far away, they may also be used when the temperatures to be measured are too high to allow safe contact.

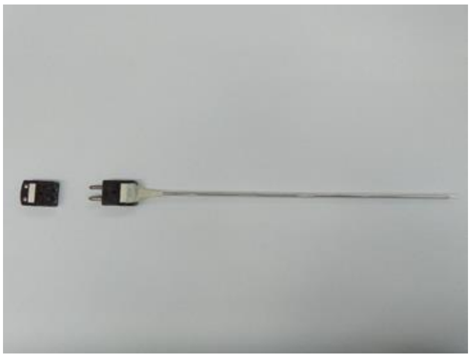

Figure 4-20 – Example of a MB1.MS4.SS16 – A temperature Sensor Omega, that uses a thermocouple.

Photographed at the Portuguese Naval Academy’s Robotics Lab

4.1.4.17. MB1.MS4.SS17 - Other (Current, Voltage, Brightness, Sound, etc).

There is such a large variety of sensors that are used in UxV that is does not make sense to have a sub-system classification for all, so while the most common do have a reference in the taxonomy, all others are grouped in the "other" sub-system class.

Current and voltage sensors are present in many electronic devices, both for energy management and for fault detection[252].

Brightness sensors (usually photoresisors or photodiods) are common both as payload for environmental sensing and as direction finders or decision aid sensors for changing navigation conditions.

Microphones (and passive hydrophones) are also used in UxV for environmental monitoring, vehicle health monitoring, and for interaction between the vehicle and humans.

4.1.5. MB1.MS5 – Navigation and Control

Every Autonomous Vehicle requires a system that ensures it can move safely towards the desired goals and can perform its mission. This system relies on a group of sensors and processors to control the platform, to direct it spatially and to perform its mission. This is sometimes called the *Guidance, Navigation and Control System* of an UxS. In RAMP we divide the Navigation & Control Main System into the Sub-System responsible for directing the vehicle spatially (Navigation), another responsible for avoiding obstacles (Collision-Avoidance), another for ensuring that the platform is functioning properly (Platform Control), and the another for guaranteeing that the global mission of the UxS is being addressed (Mission Control).

4.1.5.1. MB1.MS5.SS1 – Navigation System

The Navigation System, *stricto senso*, is responsible for deciding the trajectory the vehicle will follow (with a few exceptions introduced by the collision avoidance system). The movement objectives can be given in various ways, such as a list of waypoints, a patrol area, or an object following objective. These in turn can be given directly by the control station (pre-loaded or uploaded in real-time) or may be determined by the on-board mission control system. The way they estimate the position can also very greatly, from those that require an external navigation aid (such as a satellite constellation for GPS, radio beacons, or sound buoys for Long Base Line (LBL) navigation), those that require the UxV to emit signals (such as RADAR or laser), those require only passive observation of the environment (such as visual terrain following or visual formation movement), and those that purely internal to the vehicle (such as purely inertial systems or dead reckoning), In RAMP, we classify the Navigation Systems according to what type of sensors they use to determine the position of the vehicle[99]. We consider the following Navigation Systems:

4.1.5.1.1. MB1.MS5.SS1.1 – Inertial navigation (e.g. accelerometers)

In this category we include all navigation systems that do not require information sensed at a distance from outside the vehicle. The purely inertial systems require only accelerometers or gyroscopes, and integrate the information given by these sensors to estimate the position of the vehicle. In a broader sense, inertial systems include those that add to this estimation based on known effects of the propulsion system (dead reckoning based on motors and control effectors), and local sensors such as odometers and compasses. Inertial systems tend to have limited accuracy and tend to have a significant drift with time[253].

Sensors systems used by SS1.1: MEMS accelerometers, fiber-ring accelerometers, mechanical gyroscopes, optical gyroscopes, pressure altimeters, pitot tubes, odometers, compasses, light sensors, etc.

4.1.5.1.2. MB1.MS5.SS1.2 – Radio system navigation (e.g. GPS, LORAN -C)

This broad category of navigation system receives an external radio reference, and computes it position based on this. We include in this category all satellite-based navigation systems, generally known as GPS, but they may include systems such as GLANOSS or GALILEU besides the DoD's GPS. We also include land-based radio navigation systems such as Long Range Navigation (LORAN) -C and Tactical Air Navigation (TACAN) and various ILS, together with simple radio-beacons[254]. All these navigation systems have the limitation of working only in the air and surface (they generally do not allow underwater, underground, or in-building navigation), but provide constant and reliable updates.

GPS in particular, is ubiquitous in UAV, USV, and UGV due to its low cost, precision, and ease of use.

Sensor systems used by SS1.2: GPS, Galileu, Glanoss, LORAN -C, Omega, TACAN, radio-beacons, ILS, etc.

4.1.5.1.3. MB1.MS5.SS1.3 - Acoustic system navigation (e.g. LBL)

These navigation systems are used mainly underwater, where radio is not available. They usually rely on a set of buoys that emit an acoustic signal, and allow the vehicles to, by triangulation, compute their position. They are sometimes divided into LBL, SBL, and USBL systems. Acoustic navigation could be used for UGV and UAV, but we do not know of any implementations[255].

Sensor systems used by SS1.3: LBL, acoustic beacons, bells, etc.

4.1.5.1.4. MB1.MS5.SS1.4 – Active sensing navigation (e.g. RADAR, SONAR)

These navigation systems require the vehicle to emit signals (radio, laser, acoustic, or structured light) that interact with the environment and return a signal to the vehicle. These systems usually require a comparison between the signals returns and some type of chart, but they may also construct that chart "on the fly" in what is known as SLAM. While the comparison with a known chart gives the system a global positioning, techniques such as SLAM or DVL gives a local or relative positioning.

Sensor systems used by SS1.2: RADAR, LIDAR, SONAR, DVL, etc.

4.1.5.1.5. MB1.MS5.SS1.5 – Passive sensing navigation (e.g. visual terrain following, passive RADAR, astronomic navigation)

These navigation systems rely on data that the environment send to the vehicle. The most common are visual navigation systems that either use some type of known chart, visual targets, or build a model of the environment (SLAM, seen earlier). The signals can also come from other sources, such as passive RADARs that use signals emitted by non-navigation systems (such as satellite TV or commercial radio stations). Formation navigation, where a vehicle follows another, is usually done using vision in UxV, but can also be done using active sensing.

Sensor systems used by SS1.5: Vision, passive RADAR, etc.

4.1.5.1.6. MB1.MS5.SS1.6 – Other navigation systems

New and creative navigation systems are abundant. The use of lighthouses, while very classical is not included in any of the above categories. The use of magnetic charts, although possible to classify under SS1.5 is also a category by itself, together with thermal orientation systems.

Sensor systems used by SS1.6: Observation of lighthouses, etc.

4.1.5.2. MB1.MS5.SS2 – Collision Avoidance

The collision avoidance subsystem of navigation and control is extremely important to guarantee the safe operation of the UxV in a non-segregated (i.e. a "common") space. It has been one of the stumbling blocks for the legal acceptance of UxV, since regulators require a guarantee that the UxV will not damage or be a nuisance to other vehicles, objects, and mainly humans. Collision avoidance sub-systems should always exist, and for legal reasons there should always be a simple and clear way to audit them and perform "post-factus" analysis of their logs[256].

Collision avoidance systems should temporarily override the other navigation sub-systems when a collision is imminent but must interact with the other navigation sub-systems to minimize deviations from the planned path, and mainly to avoid getting stuck in a deadlock (such a UGV insisting in going against a wall and then backing off).

These systems usually rely heavily on onboard sensors, such as RADARs or ultra-sound proximity sensors, but may also be based on information from the ground system. In the latter case, some ground-based sensing device (such as RADARs or self-reporting systems such as Automatic Identification System (AIS)) will detect multiple vehicles and have knowledge about fixed obstacles (e.g. from a map) and direct the UxV to a safe path[257].

4.1.5.3. MB1.MS5.SS3 – Platform Control

The platform control system is responsible for managing all assets aboard the UxV, guaranteeing the safety of the platform, and ensuring its correct operation and survivability.

The platform control systems will usually perform power-on verifications, many times detecting the current configuration and state. This may have an impact on the missions that the UxV can perform, and thus on the commands it will accept or not.

During operation, the platform control system may be tasked to keep a given attitude, direction, altitude, depth, or speed, under the control of the Navigation or Collision avoidance sub-systems and using the available sensors and actuators.

In more sophisticated systems, the Platform Control sub-system is responsible for health-monitoring, overseeing potential problems such as overheating, excessive vibration, low energy, failure of systems, or even damage control.

4.1.5.4. MB1.MS5.SS4 – Mission Control

A UxS is not an end in itself, and its operation will always have an objective, or a *Mission*. In a purely remotely-operated system, the mission control will be in the ground segment, where a human or a computer mission control system will direct the actions of the UxV to accomplish its mission. Even for these remotely-systems, some basic mission control must be available aboard the UxV to deal with communication loss. In this case, the original mission is aborted, and the mission becomes a simple recovery of the UxV. The most basic mission control sub-systems will return to the launch location, loiter, or land/resurface the UxV.

As UxV gain more autonomy, most have a mission control sub-system aboard the vehicle that can guarantee the accomplishment of the designated mission without assistance from the ground segment. The mission control sub-system will usually control all other Navigation and Control Sub-systems, directing the UxV to the desired locations and controlling the payload. In a human operated vehicle, this is the function of the "mission specialist" or "weapons officer", and in some cases the payload operation takes up most of the resources. The mission control sub-system will usually be a state-machine to account for the various phases of the mission, such as launch, transit, search and acquire, engage, disengage, transit, and recover[258].

4.1.6. MB1.MS6 - Payload

Payload is every equipment that is taken on the vehicle to perform a given mission but is not part of the vehicle itself. It will many times include sensors already covered in MS3, but these can be managed as part of the sensor system or managed completely separately. In either case, the information provided by the sensors is normally treated quite differently when the sensor is used as payload (in which case there is normally a ground control payload operator to process it), and when it is used to control the platform (in which case the information is used by the Navigation & Control systems). Other payloads have nothing to do with sensors. They may be actuators (crop spraying mechanisms, lighting systems, armaments, buoy dispensing systems, etc), or simply transport systems. Due to the extremely diverse types of payloads that can be used on UxVs, we do not list specific subsystems in the RAMP taxonomy, and only list three broad categories:

- **MB1.MS6.SS1** – Sensors (including as sub-categories all those listed in MB1.MS3)
- **MB1.MS6.SS2** – Actuators (including dispensing systems, robotic arms, armaments, etc)
- **MB1.MS6.SS3** – Passive transported cargo.

4.2. Datalink Components (Main Systems - MB2.x)

As seen when discussing the Vehicle (MB1.MS2), some communication is always necessary between the vehicle and the ground segment, and possibly with other interlocutors. The Datalink Main Block concerns what is outside the vehicle and the ground segment. The different elements of the Datalink are well described in reference models such as OSI discussed earlier, but the two most important elements are the physical layer (that sets the standards for the physical electrical, acoustical, or electromagnetic signals), and the OSI datalink layer (or logical layer) that specifies the logical connection. At the physical layer, the datalink may be composed of a wire, an acoustic transmitting medium (water or air), a radio connection (through air, vacuum, or other media), each with a number of different standards.

The Datalink Main Block may have multiple communication channels. These can be categorized in many ways. One common division is uplink and downlink. Uplink are usually all communications that are sent by the ground segment, and downlink all communications sent by the vehicle. However, this can be confusing when we are dealing with UUV (since the vehicle is usually lower than the ground station) or when satellite communications are used. These names are also misleading because the uplink or downlink may actually require full-duplex (or at least half-duplex) connections. Thus, we refrain from using them. We can also categorize these links according to their function, as Control Links (for commands and reporting of the platform), and Data Links (usually for payload data). Finally, they may be categorized according to frequencies or required ranges (VHF links, UHF links, etc.).

Three of the most important variables to take in consideration when choosing a datalink are: latency (critical for online control of UxV), bit rate (especially when online reporting, such as video-streaming is required, or when the medium is particularly slow as happens in underwater communications), and error and data package loss rate.

Many different Datalink systems exist, but we shall mention only some to exemplify that this main block is:

Radio Control Datalink – In its simplest form, a radio control datalink has just an uplink to transmit controls to servos, using Pulse Width Modulation (PWM) over Very High Frequency (VHF) channels. More recent radio control datalinks use higher frequencies (in the 2.4 GHz range) using digital signals over full-duplex channels.

Wi-Fi Datalink – WiFi is a wireless local area network based on 802.11 standards of the IEEE that allows communication between all the equipment inside the network at a short range[259]. In most cases it operates in the 2.4 and 5 GHz band and it has a range of approximately 100 meters, but this can be extended to 5 km, or even more with signal boosters. Due to its low cost and wide availability, WiFi is widely used in UxS. Similar datalinks, such as WiMax, or WiFi over VHF are also being used[260].

Global System for Mobile Communications (GSM) Datalink – GSM is a cellular network mostly used by mobile phones but can be used by UxV and GCS using digital protocols for second-generation (2G), 3G, 4G, LTE, etc. These communications usually use commercial service providers[261]. The equipments themselves are cheaper than dedicated Radio Frequency (RF) datalinks and easier to use than Ethernet cables, which can be limited in some situations. However, there is usually a cost associated with each data packet sent.

Satellite Datalink – For over-the-horizon communications, satellite datalinks are a convenient alternative, but usually have a very high operating cost. They are however widely used as emergency or “watch-dog” communications, since services like SPOT, or Global Maritime Distress and Safety System (GMDSS) distress messages, which allow only short messages to be send from the vehicle to the satellite, are quite cheap.

Underwater Datalinks – Few underwater datalink protocols are widely used. However, NATO recently approved STANAG 4748 that defined JANUS as a common standard for digital underwater telephone and communication with UUVs. Another example of widely used underwater downlink (from the vehicle to the ground segment) datalinks are the emergency pingers and HiPAP.

4.3. GroundSegment Components (Main Systems - MB3.MSx)

As stated in the beginning of the chapter, the ground segment includes all support systems that are not aboard the UxV. They vary tremendously, from just a computer that writes instructions on a memory card, to a hand-held gamepad, to a full-blown multiple container system with multiple work stations (Figure 4 21)., catapults, maintenance workshops, etc. They can all be conceptually divided into 4 main systems.

Figure 4-21 - Example of a MB3.MS1 –Control Station

Photographed during the final demonstration of research project SUNNY in April 2018, São Jacinto, Portugal

4.3.1. **MB3.MS1 Control Station (GCS)**

The Control Station of the Ground Segment is the system responsible for:

- Mission Planning (SS1). Preparing and planning the mission of the UxV, wither with high-level objectives, waypoints, or other means. This may be done using a scripting language or a graphical interface.
- Vehicle Monitoring (SS2). Monitoring the vehicle during the mission, including showing its position on a map, displaying its speed, bearing, battery state, etc.[262]
- Vehicle Control (SS3). Redefining goals and behaviors of the UxV during the mission. This may vary from "flying" a UAV in a manner very similar to pilot in a manned aircraft, to a very broad effects-based approach of assigning patrol areas and rules of engagement to a USV.
- Payload Control (SS4). Controlling the payload aboard the vehicle.
- Data Processing (SS5). Receiving and processing information passed on by the UxV, mainly from the payload, but including all data gathered by the UxV.

In virtually all GCS, the Human-machine interface is of crucial importance, although it varies tremendously with how the UxS is controlled. There has been a lot of effort to make this Human-Machine interface fast, reliable, unambiguous, intuitive, and effortless for the operators. There are concerns with the ergonomy of the chairs and workstations of the operators, the use of virtual reality systems, haptic feedback, and voice-controlled interfaces[263].

For small systems, one human operator is enough to perform all tasks involved with preparing and executing a mission, but the workload on more sophisticated systems will usually require multiple people. It is very common to have a “Pilot”, which has the responsibility of monitoring and controlling the platform (and in RPAS to actually “fly” the UAV), and a “Payload Operator”, which has the responsibility of analyzing the data received from the UxV and operating its sub-systems. Additionally, there may be a “UxV Engineer” to monitor the state of the machinery itself and a “Mission Commander” to oversee the whole team and keep the focus on the mission to be accomplished[264]. This organization is strongly influenced by the standard organization aboard combat aircraft.

The data received by the CS from the instruments can be processed on-site or forwarded to a processing center via telecommunication means.

4.3.2. MB3.MS2 Communications

The communications main system of the ground segment (MB3.MS2) are quite similar to the communication systems aboard the vehicle (MB1.MS2). However, on the ground station there is usually more space and less pressure to minimize energy consumption or size. Thus, it is common for the MB3.MS2 to have more channels available, more power transmitted, larger but more sensitive antennas, directional systems, etc[265].

4.3.3. MB3.MS3 Launch and Recovery

The performance of an UxS relies greatly on the capability of launching and recovering the UxV. The procedures and devices used to do so are referred to as Launch and Recovery System (LARS), and must deal with various problems: safety to all involved during the launch and recovery period; integrity of the UxV during the process; host (i.e. platform that transports and launches the UxS) to

UxV interfaces; LARS's portability; maintenance and storage of the LARS; man-power requirements, *etc*[265].

Some vehicles require an electrical, mechanical or hydraulic platform to be launched due to their weight or size. Micro or small vehicles can be hand-launched easily, and thus don't need a physical LARS but, even in these cases, certain procedures must be observed to ensure a safe and successful operation. In almost all cases, the LARS must ensure some type of pre-launch system check, including calibration, mission plan loading and emergency backup procedures. In the case of recovery failure, many UxV have emergency localization devices (acoustic pingers, GPS trackers, etc) to ensure final recovery. When the UxV operates from another vehicle (possibly it too a UxV), the LARS can be quite challenging.

UGV's typically don't present much of challenge when it comes to launch and recovery, due to their own nature.

For a UAV, the LARS is typically a more complex system.

For rotary winged UAV, the challenge is little to none since they allow vertical take-off and landing. They can take off or land in an open field, a deck of a ship or even be retrieved by hand (small and micro UAVs), with very little man-power involved. For larger rotary winged UAV, landing on a ship's deck is like a landing a manned helicopter. Thus, a system such as the Aircraft Ship Integrated Secure and Traverse System (ASISTS) or Light Harpoon Landing Restraint System (LHLRS) can secure the UAV when it lands, using winches and grids.

Fixed-wing UAVs present a greater challenge when it comes to launching. Other than being launched by hand (small and micro UAVs), they can be launched from land, air, or from a ship which presents some additional difficulties. LARS for fixed wing UAVs can involve full runways for acceleration/deacceleration, that are almost always required for larger UAVs, or may be "zero-length" systems that do so in a very short space, although, to be accurate, they are never really "zero length". UAVs launching systems can be of different types:

- Rocket Assisted Take-off Systems have the advantages of having a small deck footprint, a small initial cost, can be prepared in advance, and require minimal infrastructures on the ship. However, they have negative effects on visual, heat and sound signatures, and rockets require special care due to fire and explosion risks and have significant costs[266].

- Bungee Cord Systems have a simple operating principle, low signature, and low cost. However, they are limited to smaller UAVs, have high initial acceleration (that rapidly decreases), and it is difficult to predict the final velocity of the UxV due to variations in elasticity.
- Hydraulic Launch Systems provide a more predictable force throughout the launch phase, providing a repeatable launch with a quick reset. They are adaptable to different UAVs and have low recurring costs. However, they have high up-front costs, and a large deck footprint[267].
- Pneumatic Launch Systems, although using a different fluid, are very similar to the Hydraulic ones, and have the same advantages and disadvantages. Pneumatic systems already exist aboard most aircraft carriers, where they are known as steam launchers, for manned aircraft[268].

UAV recovery systems can be of different types:

- Net Recovery Systems provide the advantages of being simple solutions of zero length recovery. However, they require a lot of manning, have a large deck footprint, long setup time and have a great risk of damaging the UAV (Figure 4-22)[269].

Figure 4-22 - Example of a MB3.MS3 –Net Recovery System

Being tested at the Portuguese Naval Academy soccer field

- Arresting Line Systems (using horizontal lines) provide quick, zero length recovery. Also, just like the steam launchers, they already implemented on aircraft carriers. However, they involve rough landings, are typically part of a fixed structure of the ship and suffer from ship motion.
- Skyhook Systems provide an innovative design, with zero length, and require low manning. Basically, a Skyhook is a vertical line that is caught by a hook on the tip of the wing of the UAV[270]. However, these systems have large stowage requirements, can suffer from ship motion, and involve very rough recoveries.
- Parasail Systems can be used for both launch and recovery (in the latter a simple parachute can be used), have smooth recovery, and most safety risks are moved away from the landing platform. However, these systems depend on wind conditions, and may involve the presence of a permanent winch.

For UUVs and USVs, launch and recovery can be a challenge as well, since it usually depends highly on the weather conditions that the host platform is exposed to. Davits and stern ramps are commonly used to deploy UUVs and USVs from ships. Davits are a common equipment aboard ships, and they can be of three different types: slewing, A-type frame and overhead telescopic davits. Another popular method is using a fix or movable stern ramp. These ramps are used to slide down the vehicle when launching, and to winch the vehicle when recovering.

Davits have the advantage of being interoperable with several vehicles, being widely available, and are cheaper and easier to install than stern ramps. However, Davits involve heavy lifting during some amount of time, which results in long launch and recovery times, proving an operational challenge. They also require a lot of manpower for their operation[271].

Stern ramps are a more expensive option that usually has a high impact on the ship's design. They can't usually be bought "off-the-shelf" are designed for a certain kind of vehicle, which lowers interoperability with several kinds of vessels. However, they prove to be easy to operate, they are fast, and they usually require less personnel than other systems[272].

4.3.4. **MB3.MS4 Support Equipment**

The Ground-Systems Support Equipment includes operating and maintenance manuals, consumables, first-line servicing items, tools, subsidiary equipment and transportation devices.

Operating and maintenance manuals are items which have information about operating instructions, specifications, time logs and maintenance instructions. This information allows a user to know how to setup and shutdown the system, to run operability checks, to store the history of the system, and how to replace certain modules of the system.

Tools (like test meters, battery-chargers, rigs, torque spanners, etc.) and consumables (like lubricants, cleaning material, batteries, fuel, etc.) are important items to keep close to the control station and in storage, since they guarantee the UxS' functionality.

Depending on the dimensions and weight of an UxS, its transportation can vary between portable backpacks, land vehicles, towed trailers, airplanes, small boats and ships.

5

Validation

In this chapter we will validate the proposed reference model and show that a single standard (JAUS, one of the IBBs reviewed) can be used over a wide range of UxS. During this thesis we worked on the Autoland, Seagull and GammaEX projects that were important to consolidate and validate the reference model of the UXS. We also worked on the ICARUS, smart unattended airborne sensor network for detection of vessels used for cross border crime and irregular entry (SUNNY), and cooperated with UAVision on a preliminary project, which were important to understand and solve interoperability issues, again using RAMP as the reference.

5.1. **Validation of the reference model**

We validate of the reference model by showing how it can be applied in various cases. These cases were research projects on which we worked. The experience gained in these projects allowed us to understand the issues of a reference model, and to consolidate the proposed model. The fact that in all these research projects the RAMP model makes sense and maps the most relevant components of the systems, is a validation that the model can be used in a large variety of cases. In the first project, Autoland, the system architecture was quite general, and a mapping to RAMP is trivial, but not very enriching. The second project, Seagull, was important to consolidate our view of what RAMP should be, and how important it is. In the last project, GammaEX, the system architecture was strongly based on RAMP, and so follow it very closely.

5.1.1. Autoland

The Autoland project had two partners: Portuguese Navy Research Center (CINAV) and Tekever. It was started in 2013 and finished in 2016.

One of the problems of operating an UAV's at sea is that they must be adapted to that environment. One of the major problems is the landing phase and this is the focus of the AUTOLAND project. The aim of this project was to adapt an existing UAV (AR-4) for naval uses, developing a localization and orientation system for the landing phase, a landing control system, and a retention system for landing the UAV on a ship.

With these developments the UAV could be used aboard ships and have a large impact in Navy missions in the future.

5.1.1.1. System Architecture

The system is based on a mini UAV with the characteristics that will be specified below. The communication between the aircraft and the GCS is done via data link, which is a service that Tekever AS provides and allows the transmission of real time video data. There is also a Nano GCS, which will be used in case of emergency, and a Remote Video Terminal (RVT) to monitor what can be seen from the UAV[273]. The basic architecture of the system can be seen in Figure 5-1.

Figure 5-1 - Autoland system architecture

This architecture, although very simple, maps very well to RAMP. The Autoland UAV is the MB1, and has all its Main Systems: MS1-Platform (the airframe), MS2-Communications (a 400MHz and 900MHz radio control and a 1-6 GHz radio for payload data), MS3-Power and Propulsion (with all its subsystems including the battery, a single brushless electric motor, and single push propeller), MS4-Sensors (pitot tube, GPS, accelerometers), MS5-Navigation and Control (onboard computer for navigation and control), MS6-Payload (a video camera) . In the Autoland system architecture only the MB1 (UAV), and MB1.MS6 (Payload) are explicitly represented in the figure that explains its architecture.

The Autoland Datalink is the RAMP MB2 (Datalink Main Block), but in the drawing it also represents the MB3.MS2 (Ground Segment Communications). The other part of the datalink equipment, MB1.MS2 (onboard communications) is implicit.

The Autoland NanoGCS, GCS, and RVT (Remote Video Terminal) are part of the RAMP Ground Segment (MB3) and are all examples of Control-Stations (MB3.MS1). The Autotland GCS has all the subcomponents of the RAMP MB3.MS1, in particular it has SS1 (mission planning), SS2 (vehicle monitoring), SS3 (vehicle control), SS4 (payload control), and SS5 (data processing). The Autoland NanoGCS only has SS3 (vehicle control) and this is done basically in direct Radio-Control (RC) mode. The RVT is also a GCS (MB3.MS1), but only has the SS2 (vehicle monitoring) sub-system.

The Autoland UAV has the following characteristics:

- Wingspan: 1800 mm;
- Length: 1200 mm;
- Weight: 3 Kg + 2 Kg payload;
- Cruise Airspeed: 55 Km/h;
- Autonomy: 2 h;
- GPS;
- Attitude and Heading Reference System (AHRS);
- Auto-Pilot;

5.1.1.2. **Work Packages**

The AUTOLAND project was divided in work packages which included *preliminary studies*, *technical specifications*, research and development of the *landing system*, research and development of the *navigation system*, research and development of the *retention system*, *autonomous system modifications* and finally *tests and results*. Each stage of the project will be resumed in the following sections.

5.1.1.2.1. Preliminary Studies

In the *preliminary studies*, and operational requirements were established and on overview of the state of the art in this area was made[274]. We analyzed the missions of UAVs in the maritime environment[275], with a particular emphasis on the problems with take-off and landing[276] and the communication between the UAV and the GCS[113].

The state of the art focused on the current state of ship-borne landings systems, methods for marking the landing areas, identification and location of targets, among others[277]. From this analysis we concluded that the UASs that existed on the market needed to be improved and their systems adapted to be used in the maritime environment. Issues such as storage in confined spaces, the impact on the crew and the ship, and the equipment related to launching and landing the UAV needed to be addressed, and existing solutions improved.

5.1.1.2.2. Technical Specifications

This work package consisted in defining the technical specifications of the system, which was divided in: system reference model, navigation system requirements, landing system requirements, and finally retention system requirements.

The objective of the work package of the *system reference model* was splitting it into physical and functional components:

- GCS - Coordination and operation of the whole system UAV;
- UAV - System that contains all the sensors, to make the flight autonomously. In this case it must have mechanisms for location of beacons of the restraint system;

- Nano GCS – A system independent of the GCS, created to give security conditions. It can be activated by the operator of the GCS so that a local operator can control the final landing phase of the UAV;
- Communications System – This system includes radios, antennas and supporting structures;
- Antennas – The antennas were developed and tested in order to have a 360º coverage, a range of up to 3 km in altitude and a 20 km horizontal coverage radius (Figure 5-2).

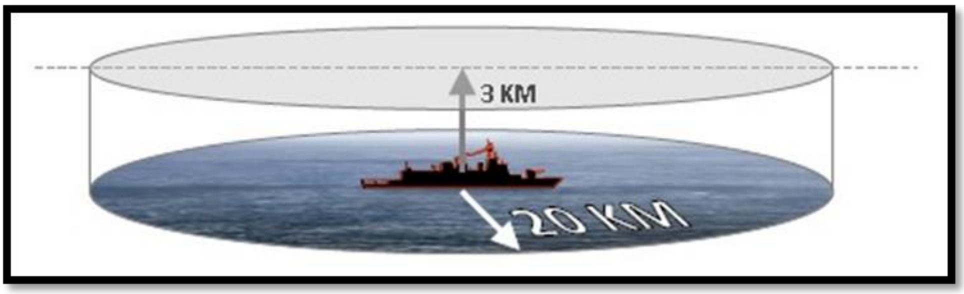

Figure 5-2 - Communications coverage

The landing system requirements technical specifications included the location of the light beacons and the controller, the accuracy required of the identification and location systems, sampling frequencies and energy consumption of the beacons and sensors, amongst others.

For the *navigation system requirements*, the maximum error in the waypoint positioning was defined, together with condition to switch amongst different phases such as launch, transit, loiter, or land[278].

The requirements for the *retention system requirements* defined aspects such as the maximum weight, the robustness of the retaining frame, the damping factor, amongst others.

5.1.1.2.3. Research and Development of the Landing System

This work package consisted in developing the guidance system for landing, following the specified requirements. This package was divided into the following tasks: *choice of the set of beacons and sensors* to be installed; *beacon identification algorithms*; *target location algorithms*; *UAV pose detection algorithms* in relation to the target; and *landing controller*.

In the *choice of the set of beacons and sensors* research task two approaches were considered: the first consists of using infrared and radiofrequency beacons aboard the UAV, which are detected by the ship. The ship-borne system then transmits the relative position to the UAV. The advantage of this approach is that processing is done mainly aboard the ship. The second approach is the placement of infrared or radiofrequency beacons on the ship, which are detected by the UAV that then processes the data.

In the *beacon identification algorithms* task, the algorithms that allow the identification of beacons by sensors were developed and implemented. In this process several algorithms were developed, with images and data collected from flight tests[279].

The *target location algorithms* use data provided in the previous task to determine the relative location of the UAV and the retention system. Various approaches were tried and in the end a system that uses Efficient Perspective n Point (EPnP) [280]was used, thanks to its fast processing and high accuracy.

For *UAV pose detection algorithms* several approaches were attempted, including stereoscopic vision using two cameras to observe different angles and extract a 3D scene by identifying points of interest (Figure 5-3). In the end, one of our colleagues developed a monocular system using a point cloud approach that, by requiring a single camera, is easier do maintain and calibrate[281].

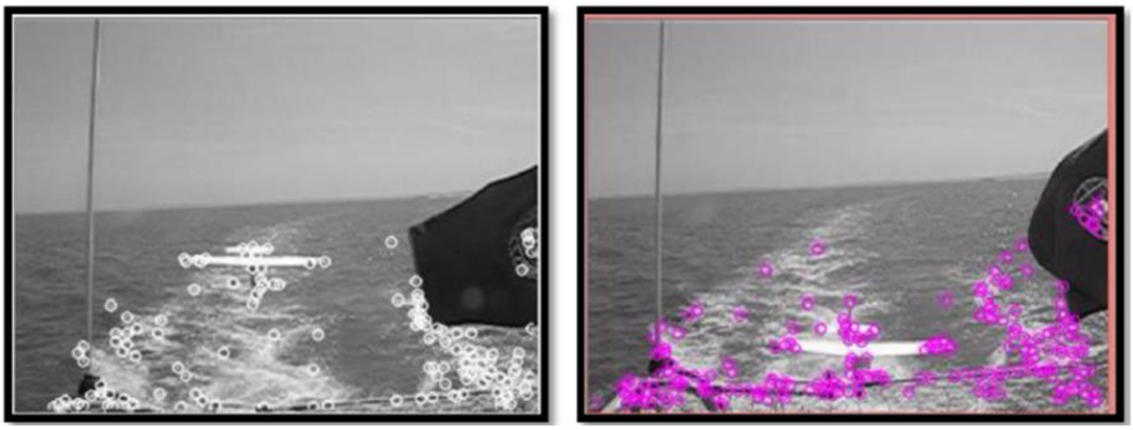

Figure 5-3 - Tests made to get the interest points in the detection system by the vessel

The last task was to develop the UAV's *landing controller*. This controller uses information from the previous tasks and information from the ship sensors to guide the UAV into the retention system[282].

5.1.1.2.4. Research and Development of the Navigation System

The purpose of this work package was to develop and implement the system to pre-position the UAV for landing on the ship. It had three tasks: *navigation system*; *landing mode switching* algorithms; and *supervision and decision system*.

The first task, *navigation system*, takes into account weather conditions, air/sea traffic conditions, prohibited areas, and ship maneuvering, to plot an approach route for the UAV[283].

The second task. *landing mode switching* algorithms, uses information from various sensors to determine when the UAV is ready to initiate the final approach. An example of a situation where this switching can be done is presented in Figure 5-4. Even after the UAV has gone into final approach mode, this system continues to monitor what is going on so as to decide to go back to the previous mode (failed landing)[284].

Figure 5-4 - Beacons capture test

The final task *supervision and decision system*, responds to breakdowns or mechanical failures throughout the process, to ensure the safety of the UAV and the personnel involved.

5.1.1.2.5. Research and Development of the Retention System

The purpose of this work package was to research and develop the retention system on board the ship. This system should stop the UAV in a safe manner so that there is no damage to the structure of the UAV or of the ship. This task had two parts: developing the retaining structure; and developing the cushioning and protection system.

Several types of retaining systems were studied, including the use of a horizontal cable, a vertical cable, and a net[269]. After a few tests, we opted for a traditional net, with a slant of approximately 45º as shown in Figure 5-5.

Figure 5-5 - Net retention test

The second part was the development of the cushioning and protection system. Since the system selected was a net, this part consisted in choosing the right materials for the net, to have the necessary elasticity/rigidity.

5.1.1.2.6. Autonomous System Modifications

The purpose of this work package was to modify the existing autonomous system (AR-4) to be compatible with the marine environment and the solutions developed previously. The main change to the GCS was the provision that the take-off and landing locations changed constantly (due to ship movement). The UAV needed more changes, since it had to be all waterproofed, detachable parts (that existed in the land version) had to be secured (Figure 5-6), the whole structure had to be reinforced to withstand the violent landing phase, and safety control parameters had to be adjusted to allow high-angle final approaches.

Figure 5-6 - UAV adaptations

5.1.1.2.7. Tests and Results

Several tests were made using a piecewise approach.

First, the various components were integrated in the UAV and the whole system was tested in the laboratory to make sure everything was working normally, and all possible calibrations were made.

The second set of tests were basic flight tests. The airworthiness was verified, and basic flight characteristics, such as speed, path following, etc. were tested.

After this six-fundamental project-related tests were performed. The first consisted in estimating the position of the UAV in relation to a fixed beacon (this test was carried out on land). The second test was identical to the first but this time the beacon was moving. The third test was conducted in the simulator and the purpose was to test the control algorithms and the path planning. The fourth test was designed to check the landing of the UAV ashore (with a fixed net). The fifth was also like the previous one but this time with a moving net. The sixth and final test was the real test aboard a ship, to test the entire system. This final test confirmed that the UAV can land safely in a ship (Figure 5-7).

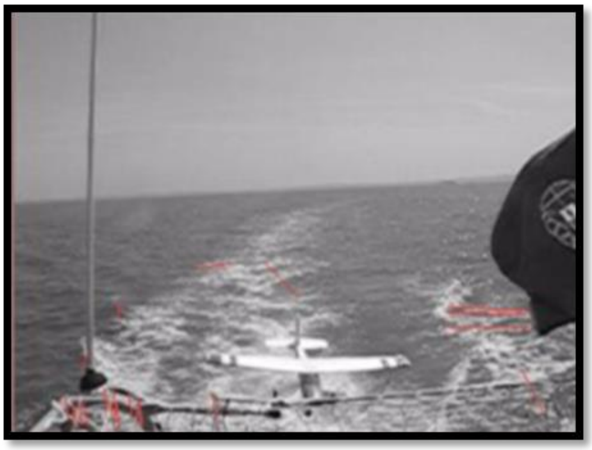

Figure 5-7 - Final landing tests

5.1.1.3. **Conclusions**

The project was completed with a successfully landing ratio of 80%, which comparing with other systems in market was very good. The objectives were exceeded, and this system had very good market reactions at an international level. The AUTOLAND markets are the military and security forces, therefore the presence of Portuguese Navy Academy and the Portuguese Navy was decisive for the credibility of the system.

5.1.2. **Seagull**

The Seagull project had five partners: CINAV, CRITICAL Software (CSW), FEUP, Portuguese Air Force (Air Force Academy Research Center) and University of Lisbon (ISR/IST). It started in 2013 and finished in 2016. The UAVs used in this project were built by the Air Force Academy Research Center. They have an autonomy of 8 hours and take-off weight of 25 kg with a payload of 10 kg.

SEAGULL’s objective is to develop intelligent systems and equipment, like optic and infrared camera systems, to integrate in UAVs that already exist in the market, in order to improve maritime situational awareness. This requires the development of detection, classification, identification and target following algorithms (for example for oil spills, shipwrecks, amongst others)[117], as well as algorithms to recognize behavioral patterns (for example high speed vessels or non-typical navigation patterns) or monitor environmental status.

5.1.2.1. System Architecture

The Seagull's System Architecture has an open system approach, in order to maximize the number of standard components, protocols and interfaces, and simplify interoperability of equipment and software, possibly providing multiple redundant systems for the same function[285]. This architecture is presented in Figure 5-8.

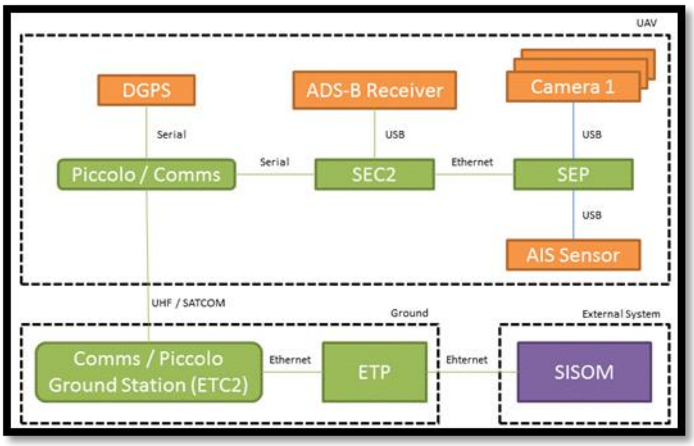

Figure 5-8 - Seagull System Architecture

This architecture has much more detail than the one discussed previously, and it yet it still maps very well to RAMP.

The Seagull UAV (top rectangle in the figure) is the MB1 (Vehicle Main Block), that naturally has an airframe, known as MB1.MS1(Platform) in RAMP.

The main component of the Seagull UAV is a commercial *Piccolo* micro controller which implements the autopilot, therefore controlling the UAV, and implementing MB1.MS5.SS3 (Platform Control) and MB1.MS5 (Navigation and Control), namely taking care of MB1.MS5.SS1.1 (Inertial Navigation), MB1.MS5.SS1.2 (Radio System Navigation). This device is connected to a Differential GPS (DGPS), which is a GPS with improved positioning and localization precision, even though Piccolo already has an internal GPS system, air data and inertial sensors. This DGPS is a subsystem in RAMP, namely the MB1.MS4.SS2 (sensors, radio-position).

The Piccolo is also connected to a *Sistema Embebido de Comando e Contolo* (SEC2, that derives from the initials of the Portuguese name, like all components of Seagull), that in turn is connected to two other Seagull Systems. The first is the

Automatic Dependent Surveillance–Broadcast (ADS – B) receiver, to get real-time information about air traffic to avoid the proximity to other aircrafts. The other system is the *Sistema Embebido de Payload* (SEP). The SEP is the computer that deals with the payload, guaranteeing target tracking. The SEP receives position data from the SEC2 and is connected to an AIS receiver (to identify targets and traffic) and three cameras, operating in thermal, near infrared and visible spectrums.

The SEC2, like the Piccolo, is part of MB1.MS5 (Navigation & Control), but has different sub-tasks. Within MB1.MS5 (Navigation and control) this module is responsible for MB1.MS5.SS2 (Navigation & Control – Collision avoidance), and MB1.MS5.SS4 (Navigation & Control – Mission Control). To deal with MB1.MS5.SS2 (Navigation & Control – Collision avoidance) it uses information from the ADS-B which in RAMP is MB1.MS4.SS11 (Sensors – Surrounding Panorama). The SEP, in RAMP, is MB1.MS6 (Payload). The SEP is connected to the AIS receiver, that in this case is used as payload, this RAMP MB1.MS6.SS1 (Payload-Sensor). The same AIS receiver, although used as payload (it is not essential for navigation or control), is a sensor and classified as MB1.MS4.SS11 (Sensors – Surrounding Panorama). All three cameras connected to the SEP are payload sensors (MB1.MS6.SS1) and as sensors are MB1.MS4.SS1 (Sensors – Image).

The Piccolo autopilot also connects to a data link radio that allows the communication with the *Estação Terrestre Comando e Controlo* (ETC2). The data link is clearly MB2 in RAMP, while the ETC2 is MB3.MS1 (Ground Segment – Control Station), being capable of SS1 (mission planning), SS2 (vehicle monitoring), and SS3 (vehicle control).

This ETC2 station has communication via Ethernet with the *Estação Terrestre de Payload* (ETP), which in turn communicates with the maritime operations support and information system (SISOM).

The ETP, in RAMP, is responsible for MB3.MS1.SS4 (Ground Segment, Control Station, Payload Control). As contemplated in RAMP, the ETP (part of the Ground Segment) has communications to external systems, in this case the SISOM.

The electrical power of the system is given by a generator coupled to the main engine, which uses internal combustion. In RAMP, the Seagull has: MB1.SS3.SS1.1 (Energy source – Combustion Fuel), that is the fuel tank; MB1.SS3.SS1.2 (Energy source – Battery Based System) that are the batteries that have their own management board; MB1.MS3.SS2.3 (Energy Transformer – Others) that is the generator; MB1.MS3.SS3.1 (Power Plant-Reciprocating Piston Engine) that is the main engine; MB1.MS3.SS5 (Propulsion Effector) which is the propeller; and MB1.MS3.SS6 (Control Effector) which are the control surfaces (ailerons and tail rudder).

5.1.2.2. **Work Packages**

The SEAGULL project was divided into 4 work packages: *primary studies*, *technical specifications*, *development and implementation* of the equipment and finally *tests and results*. These stages will be specified in the next section.

5.1.2.2.1. Primary Studies

In *primary studies* we reviewed the current situation relative to all necessary subsystems of the UAV, such as navigation, on-board processing systems, algorithms for image analysis, detection, classification and tracking of marine vehicles[286]. This task was also responsible for defining the concept of operations and operational requirements, that could be used to derive the technical specifications[113].

5.1.2.2.2. Technical Specifications

The technical specifications for Seagull defined the sensors that should be used, the processing hardware, the characteristics of the communication systems, and the various components that system should have. At this stage, a reference architecture such as RAMP would have been very useful to organize ideas and decouple related work. The separation between the Piccolo, the ETC2 and ETP and their functions, for example, was derived empirically after successive iterations, and the mapping to RAMP (that was influenced by it) came after the project had ended.

5.1.2.2.3. Development and Implementation of the Equipment

The development activity consisted in the implementation of the various components (Figure 5-9) made according to the specifications developed in the previous work package. Some parts were greatly simplified by using existing hardware and software, with only minor adaptations. The autopilot, for example, is the commercial Piccolo, that required only some parameter settings. Other parts required more work, such as the collision avoidance system, that is basically non-existent in the market, and thus was developed from scratch[287]. The algorithms were first tested in a high-level language (Matlab) and were later rewritten in C ++ to be integrated into the project's software architecture[288]. The software developed for the UAV (MB1 in RAMP) ran under ROS on Intel-powered boards.

.

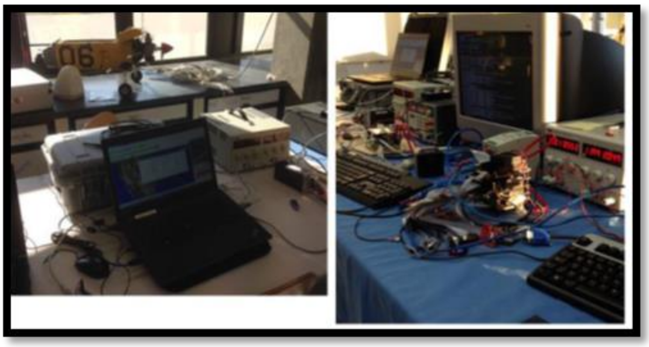

Figure 5-9 - Implementation of the components

The ground segment should have interfaced with a SISOM, which in this case was the Navy’s Maritime Situational Awareness system (Oversee), but that part was not accomplished for lack of time and funds.

5.1.2.2.4. Tests and Results

The purpose of this activity was to make the verification and validation of the various components and algorithms implemented in the different phases of the project. They began with simple tests conducted in the laboratory to assert that all components were working properly (Figure 5-10), proceeded to basic flight trials over land and then over the sea (Figure 5-11), and ended with tests in an operational scenario.

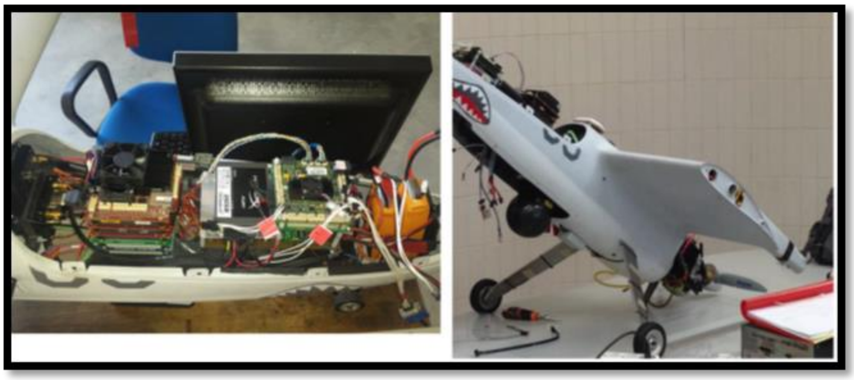

Figure 5-10 - Laboratory experiments

Figure 5-11 - Experiments and flight tests

The laboratory tests were conducted at the Air Force Academy (to test final integration), at the Navy's Damage Control School (to test the effectiveness of sensors), and Lisbon University. The basic flight tests were conducted first at OTA air base, and then at Santa Cruz Airfield that has easy access to the ocean. The final tests were conducted in the Algarve, from the Alvor airfield, and involved tracking a Navy Patrol Boat and simulated oil spills (using fish oil).

5.1.2.3. Conclusions

The objectives of this project were to develop an intelligent system that would provide an UAV with capabilities to identify and track targets, recognize behavioral patterns, monitor environmental parameters and to avoid other vehicles. The main advantage was that the prototype was tested in a very close to real operational environment. The difficulties were mostly bureaucratic, but also operational such as in the access to a testing platform, the construction of the detection and collision avoidance system, etc. Nevertheless, the project was successful, and contributed significantly to increase the skills of the consortium members.

5.1.3. **GammaEX**

The GammaEX project had five partners: CINAV, I-SKYEX, ISQ, The Portuguese Army Academy Research Center (CINAMIL) and University of Lisbon ITN/IST. It started in 2015 and finished in 2018.

The GammaEx project aims development remotely piloted aerial system capable of operating in dangerous chemical or radioactive environments, transporting sensors to detect and map those dangers. It can thus be used in military and civilian missions such as response to natural or manmade catastrophes, performing reconnaissance of nuclear, radiological and chemical agents, minimizing human intervention on CBRN operations, and therefore reducing the risk of human casualties. The UAV used was developed by I-SKYEX, and is a tricopter named M6[90].

5.1.3.1. **System Architecture**

The M6 System Architecture is composed of: a sensors module, a navigation and control module, power module, payload module, a communications module, Datalink and a Command & Control module (Figure 5-12).

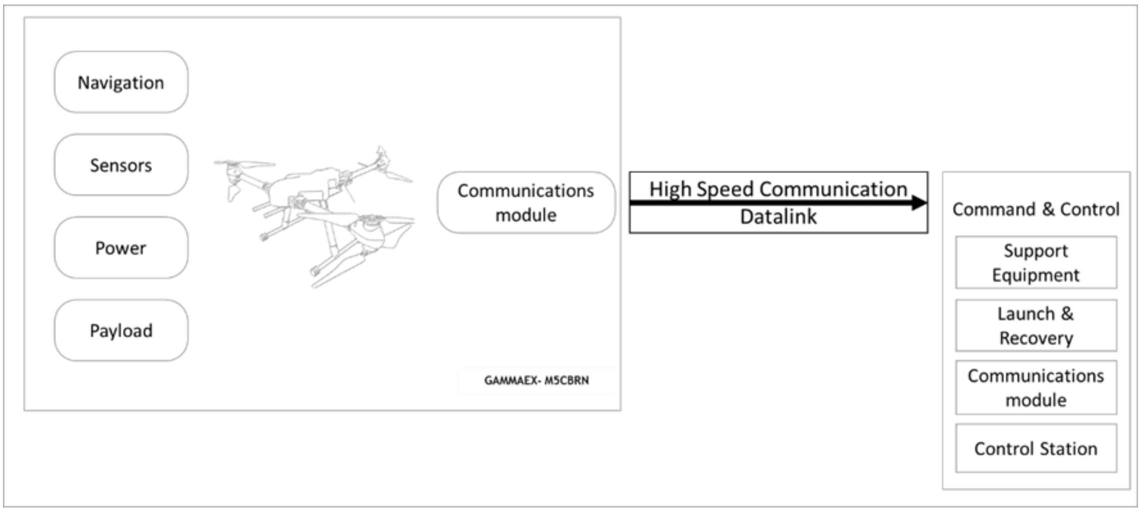

Figure 5-12 - GammaEx System Architecture

The system architecture used in this project mimics almost exactly the RAMP model because we were at a rather advanced stage of its development when we started the project. This served as an additional validation of the benefits of having a reference model: the development was streamlined and there was little doubt in defining the modules of the system.

5.1.3.1.1. Vehicle (MB1)

The vehicle used (MB1 in RAMP) has a *platform* (MB1.MS1) that is a Y-shaped structure with vertical propeller in the extremities. A normal multirotor configuration with 4, 6 or 8 thrusters has constraints due to vortex flow which inhibit the entry of suspended particles into the chemical detector. The configuration now adopted results from the analytical studies (continuity and Navier-Stokes equations) and experimental results. The direction of rotation of the propellers pulls the flow (current lines) into the chemical detector, improving efficiency.

The on-board *communications* (MB1.MS2) is a Radio-Control transceiver (a 2.4 GHz Digital RC), an 833 MHz bi-directional radio, and a 3G/4G system.

The *power and propulsion* system (MB1.MS3) uses an energy source (MB1.MS3.SS1) that is a Battery (MB1.MS3.SS1.2) using lithium-ion technology (Figure 5-13), a powerplant (MB1.MS3.SS3) that is composed of 6 Electric Motors (MB1.MS3.SS3.5), that are brushless DC motors (Figure 5-14), two of them on each extremity of the Y structure. The Propulsion effectors (MB1.MS3.SS5) are 2 blade propellers (MB1.MS3.SS5.1), that are mounted in a push-pull configuration (i.e. on opposite sides) in the extremities of the Y structure.

Figure 5-13 - Batteries

Figure 5-14 - Brushless DC motor

The powerplant does not need an Energy Transformer (MB1.MS3.SS2) because electrical power goes directly from the batteries to the motors, or Mechanical Coupling (MB1.MS3.SS4) because the motor speed can be the same as the propeller speed, or Control Effector (MB1.MS3.SS6), because control is achieved by varying the speed of each pair of motors. Therefore, the RAMP model can be used to check if all components necessary for power and propulsion are accounted for.

The *sensor* system (MB1.MS4) of GammaEX has various subcomponents, used both for the vehicle navigation and control and as payload.

It has an *image* sensor (MB1.MS4.SS1), that is a simple camera (Figure 5-15). This is used as payload (MB1.MS6.SS1), but the information can be used by the human in the GCS (MB3.MS1) to control the vehicle.

Figure 5-15 - Camera

It has a *radio-positioning* sensor (MB1.MS4.SS2) that is a GPS antenna connected to the navigation system (MB1.MS5.SS1.2).

It has a *speed* sensor (MB1.MS4.SS4) that is a pitot tube.

It has a *pose-estimation* sensor (MB1.MS4.SS6), composed of MEMS accelerometers, and an *altitude* sensor (MB1.MS4.SS8) that is a pressure sensor, both physically integrated in the platform control system (MB1.MS5.SS3).

It has a *surrounding panorama* sensor (MB1.MS4.SS11) composed of a LIDAR, that is used as payload (MB1.MS6.SS1). This sensor can also be used as a *distance-to-object* sensor (MB1.MS4.SS12).

It has an *energy-level* sensor (MB1.MS4.SS14) that is part of the battery management system, but this is used basically as payload since it does not interfere with platform control.

It has a series of *environment-parameter* sensors (MB1.MS4.SS15) that constitute the *raison d'etre* of the whole system, and their development was a a significant part of the project. A module was developed (Figure 5-16) that houses the different chemical sensors used. This module can house up to 3 sensors, and in this project, we used sensors for oxygen (O2), hydrogen sulphide (H2S), and carbon monoxide (CO). A radiological sensor was also necessary and the commercial RadEye SPRD, from Thermo Scientific was chosen (Figure 5-17).

Figure 5-16 - Chemical sensor module

Figure 5-17 - Thermo Scientific RadEye SPRD Radiologic sensor

This covers all sensors necessary both for the vehicle control and for payload.

The *navigation and control* system (MB1.MS5) is basically a open-source based autopilot, named PixHawk[289], running under Linux on an Intel-Based board. It's *navigation system* (MB1.MS5.SS1) uses radio-navigation (MB1.MS5.SS1.2) based on GPS data, receiving target way-points, but direct access to *platform control* (MB1.MS5.SS3) from the GCS is possible, overriding the navigation system (in RC mode). *Platform control* (MB1.MS5.SS3) is usually performed by the PixHawk, but as stated can be overridden by the GCS. The other navigation and control sub-systems described in RAMP are not present, since this system does not have *collision avoidance* (MB1.MS5.SS2) and *mission control* (MB1.MS5.SS4) is done from the GCS (the system is not very autonomous and relies on the remote pilot).

The *payload* (MB1.MS6) consists of the multiple environmental sensors described above, together with the LIDAR, which are all basically just sensors (MB1.MS6.SS1). Thus, there are no actuators or cargo (MB1.MS6.SS2 and SS3).

The *datalink* (MB2) consists of a 2.4 GHz Digital RC system, an 833 MHz datalink, and a 3G/4G (mobile phone) system using a commercial service provider.

The *GroundSegment* (MB3) does not need a special launch and recovery system (MB3.MS3) because the vehicle is a rotary wing system that can be launched by hand or from any flat surface and can land on any more or less flat surface. It does however have two control stations.

The main *Control Station* (MB3.MS1) runs on a standard PC (Figure 5-18), under Linux, and has all the subsystems described in RAMP. While most software was derived from the open source PixHawk[290] system, the *data processing* (MB3.MS1.SS5) was entirely developed by the project to account for its specific purpose (detect and map dangerous chemical and radiological agents), providing a user-friendly man-machine interface(Figure 5-19). The real time generation of heatmaps with chemical and radiation provides improved awareness of the threat.

Figure 5-18 - Ground Station with waterproof case

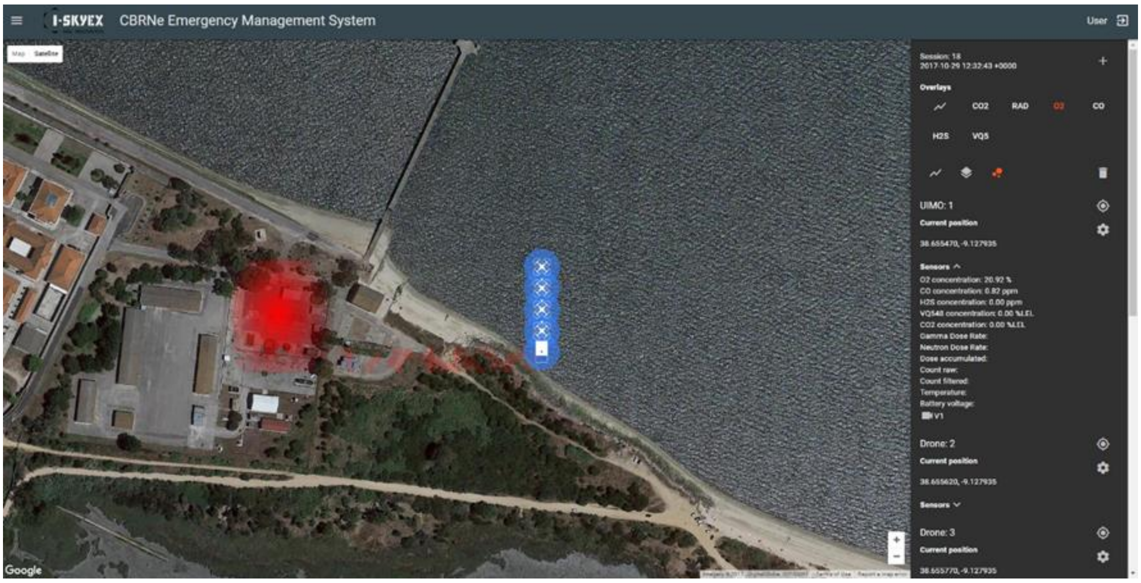

Figure 5-19 - Emergency Management System vitalization tools of the GCS

The secondary control station (MS3.MS1) is a standard RC controller that overrides the navigation and control system, allowing an operator to take over control of the vehicle.

The ground segment *communications* (MB3.MS2) is composed of a 2.4 GHz and an 833MHz transceiver, and a 3G/4G USB Communication Pen that interfaces to the commercial service provider (Figure 5-20).

Figure 5-20 - Ground Station Communication Equipment

The support equipment (MB3.MS4) is comprised of two high-quality plastic cases, one to house the ground segment components (Figure 5-21) and another for the vehicle.

Figure 5-21 - Case with Accessories

5.1.3.2. Work packages

The GammaEx project was divided in 7 work packages: project management, requirements identification and conceptualization, RPAS project and development, ATEX certification, validation of sensors integration, concept demonstration and validation, and dissemination and exploration. Each work package of the project will be resumed in the following sections.

5.1.3.2.1. Project Management

In this work package, the objective was to overview the entire project, controlling the other activities.

5.1.3.2.2. Requirements Identification and Conceptualization

The objectives of this work package were: to identify the best radiological and chemical sensors for the project; to determine the operational and functional requirements; and do define use-case scenarios[291].

The chemical sensors found didn't meet all the requirements. Because of that an electronic board that would accommodate three electrochemical sensors was developed from scratch. The software developed had two main functions: provide a driver for the sensor interfaces and provide visualization tools for the operator.

5.1.3.2.3. Remotely Piloted Aircraft System Project and Development

The objectives of this package were developing the UAV itself, the sensor integration, and the GCS.

5.1.3.2.4. Atmosphere Explosive Certification

The objective of this work package was to obtain Atmosphere Explosive (ATEX) certification. The solution adopted considered a zone 2 of explosive atmosphere and Group II category 3 certification which doesn't make compulsory a certification by a certifying entity[292]. With this, I-SKYEX made the auto certification according to the directive 2014/32/EU[112].

All IEC 60079-00 and IEC 60079-15 norms were evaluated and complied.

5.1.3.2.5. Validation of Sensors Integration

The objective of this work package to validate the effectiveness of Chemical and Radiological sensors. Tests were performed first in the laboratories of each partner, and later at the Navy Damage Control School.

5.1.3.2.6. Concept Demonstration and Validation

This package objectives were to validate the project in a Navy scenario and Army scenario and to do risk preliminary analysis.

According to the operational requirements of the GammaEx project, validation scenarios were created with the purpose of testing and validating the UAV prototype's operation inserted in CBRN, and to present and obtain the approval of the prototype from the Portuguese Ministry of Defence. These tests served, as well, to detect weaknesses of the UAV prototype and to further develop and enhance it.

The final test took place in the Damage Control School in the Lisbon Naval Base. The Army scenario was used to test an unintentional chemical threat (accidental leakage), and an intentional chemical threat (terrorist attack) in Alfeite's Port. The Navy scenario was used to test an unintentional radiation threat in the interior and exterior of a Navy ship.

5.1.3.2.7. Dissemination and Exploration

The objectives of this work package were general project dissemination, elaborating the users' manuals, international exposure at conferences and trade shows, and the preparation of proposal for a follow-on project.

5.1.3.3. Conclusions

This project proved that a UAV can be used in dangerous environments (including explosive atmospheres) and help in the response to catastrophes, such as those that occurred in Bhopal, Fukushima or recently in Tianjin. The ATEX certification of the UAV was an important requirement to operate in unknown CBRN scenarios, where nothing is known about the nature of the agent released. Finally, we believe that in the near future, UAV will be an important and necessary response tool to CBRN crisis scenarios, keeping the first respondents out of harm's way and saving lives.

5.2. Validation that a single IBB can cover a broad range of vehicles (various USV, UAV, and UGV) and conversion between IBBs

In this sub-chapter we will validate another important hypothesis of this thesis: that a single IBB can be applied to a wide range of UxS to achieve interoperability. The existence of a reference model like RAMP provides a common ground for discussing issues concerning UxS, but that does necessarily imply that we can use the same IBB over all possible UxS. There are many, and very valid, reasons to have multiple IBBs for different UxS, and interoperability may be achieved at very different levels. In one of the projects on which we worked that involved multiple heterogeneous UxS, the FP7 SUNNY Project, it was decided from the onset that due to the specificities of the vehicles, each would have its own GCS and Datalink, and use its own system. Interoperability in SUNNY is achieved only at a rather high level, when all UxS send information to a unified information console. Even there, there has to be a standard for that information exchange, but many details of each UxS are lost. Another problem is that the tasking capabilities of that unified console are limited, since the details of each system are largely hidden from it. Finally, with this solution, each UxS has to have a gateway to translate information to and from the unified console. As we discussed in the introduction, this is not a good solution. In another project that we worked on, ICARUS, the approach was radically different, and followed the philosophy defended in this thesis. A considerable effort was made to find an IBB that could be as general as possible, to cover many different UxS, and as complete as possible, so that each UxV could be controlled using only that IBB. Despite some initial resistance due to the very different characteristics of UAV, USV, and UGV, all partners agreed to have their vehicles compliant with JAUS, and in the end everything worked without flaws.

Another validation of the ideas defended in this thesis came from work done in cooperation with the UAVision company, with whom we developed a STANAG 4586 - MAVLink gateway. The development of a gateway, by itself, has nothing new, but it allowed us to further comprehend the issues related to conversions, the inevitable inefficiency of the process, and thus the advantages of using a single standard.

We will now present the ICARUS project and show how interoperability was achieved and discuss the STANAG 4586 – MAVLink gateway.

5.2.1. ICARUS and the use of a single IBB for heterogenous vehicles

The European project Integrated Components for Assisted Rescue and Unmanned Search Operations (ICARUS) started in 2012 with a large community of participants composed of 24 partners, 10 countries, 2 end-users (including the Portuguese Navy), 3 large industrials and the NATO Centre for Maritime Research and Experimentation (CMRE). The aim of the project was to search for human survivors in the event of a large crisis, such as an earthquake or a terrorist attack. Search and rescue (SAR) operations in these events are often very dangerous and put at risk many human lives. ICARUS was a large project, with 8 different Unmanned Vehicles (UxVs), and a budget of 17.5 million euros.

This project uses various UxVs, in particular USVs, UAVs and UGVs, equipped with SAR tools in order to provide situational awareness and to assist in the victim detection. In order to do so, these vehicles must be coordinated as a team and they need to communicate with each other. These vehicles are then integrated in a C4I system where they are controlled by human crisis managers. Therefore, there was a need to have a standard interface that would allow communications between the station and all the vehicles, maximizing the efforts, sharing data, intelligence and resources. Another objective of having interoperability is to facilitate the compatibility of platforms and Command, Control and Intelligence (C2I) systems with future improvements or updates to other systems, and to allow other providers to contribute with different systems, in an open standards environment that promotes competition and efficiency[293]. To achieve success, the requirements and following test scenarios were created according to the ICARUS project main objectives:

- Development of a light sensor capable of detecting human beings;
- Development of cooperative UAV tools for unmanned SAR;
- Development of cooperative UGV tools for unmanned SAR;
- Development of cooperative USV tools for unmanned SAR;

- Heterogeneous robot collaboration between unmanned SAR devices;
- Development of a self-organizing cognitive wireless communication network, ensuring network interoperability;
- Integration of unmanned SAR tools in the C4I systems of the human SAR forces;
- Development of a training and support system of the developed unmanned SAR for the human SAR teams;
- Communication and dissemination of project results.

ICARUS architecture is designed with a hierarchical view for data exchange and operator's missions and responsibilities. The head of this hierarchy is the Mission Planning and Control System (MPCS), which is located nearby the On-Site Operations and Coordination Center (OSOCC). This coordination center creates the mission plan for the MPCS system. After this, updates are sent to the Robot Command and Control (R2C) tools, which are located nearby the mission area. This R2C tools are responsible for the coordination of the ICARUS robot teams. These teams are composed by UAV, USV and UGV. Thus, these vehicles form a heterogeneous fleet, ideal for crisis incidents, which are complex situations and require different capacities.

The coordination of the fleet of vehicles is done by the ICARUS R2C. This project followed a loose coordination strategy. This way, once a team or sector are given to a robot, its missions and objectives are planned from that team's (or sector's) R2C.

Each vehicle has a "role", that defines the systems behavior. These roles are defined by the C2I. Example of roles can be "scouting", in order to explore an area or route, or "search", in order to find known victims. To perform those roles, the systems must perform a number of "tasks". Examples of tasks can be "launch", "recover" or "move to a waypoint". Finally, mission plans are created based on different roles, tasks and responsibilities.

The LOI used in the ICARUS project were based on STANAG 4586. This is important because it provides information about the level of control that a user has over the vehicle, payload or both. There are five levels of interoperability:

- Level 1 – Indirect receipt/transmission of UxV metadata and payload data;
- Level 2 – Direct receipt/transmission of UxV metadata and payload data;
- Level 3 – Control and monitoring of the UxV payload, not the unit, in addition to level 2;
- Level 4 – Control and monitoring of the UxV without launch and recovery;
- Level 5 – Control and monitoring of the UxV including launch and recovery.

These LOIs are changed through the standard interface. Figure 5-22 displays various levels of interoperability used within ICARUS.

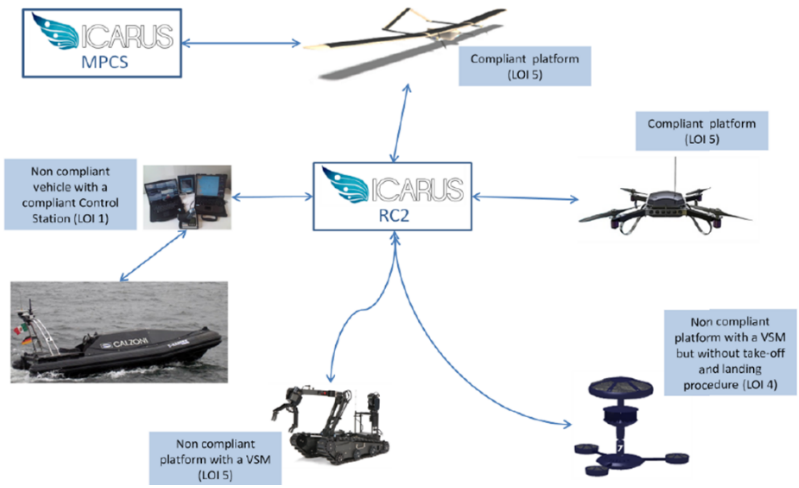

Figure 5-22 - Examples of ICARUS Levels of Interoperability

According to the ICARUS concept, there is a need for a synergy between UAVs, UGVs and USVs. This synergy allows different vehicles to perform tasks together, being able to share data, intelligence and resources, as well as to be more compatible with different control systems and stations. As such, ICARUS implemented its own standard interface to develop interoperability. The concept of a common standard of interoperability results in a reduction of integration time and development costs and in a smaller impact upon the integration of new sub systems. Following the concept behind standards, the ICARUS standard is built upon a pre-existing one (JAUS), with the necessary additions, avoiding the temptation to develop a new optimized one that would just add to the panoply of unused standards.

JAUS was chosen because it fulfilled quite well the project needs, providing services that are viable for multi-air, ground and sea vehicle operations, and has been proven to work well with in a large multi-system scenario. However other standards were still used in this project at the platform level, as will be explained later.

JAUS is a SOA, with a taxonomy that includes systems, subsystems, nodes and components[294]. An ICARUS team is considered a system, each vehicle is a subsystem with a single node and a node is composed by many components, such as cameras or lasers. Figure 5-23 describes the ICARUS JAUS topology.

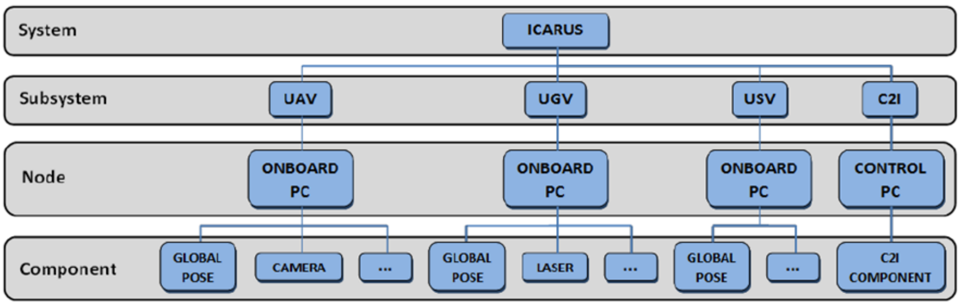

Figure 5-23 - ICARUS JAUS topology

In the ICARUS project there were lots of vehicles in the air, ground and maritime environments. Many of these vehicles don't have JAUS implemented as a native protocol, because there were developed separately and there are lots of communication methods in the marked. Therefore, adapters had to be developed for this project. Some examples are STANAG-JAUS bridges or ROS-JAUS, as can be seen Figure 5-24[115].

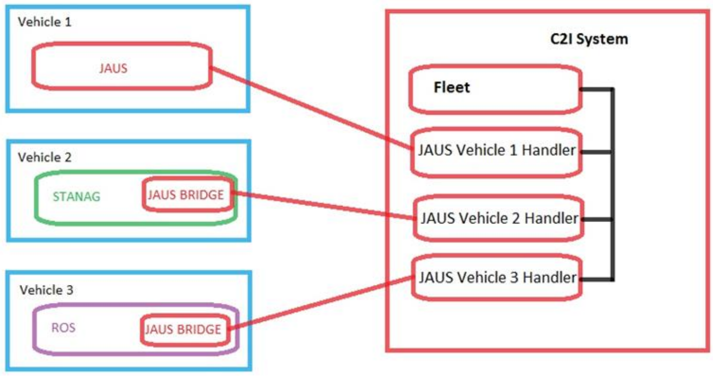

Figure 5-24 - Example of implementation using JAUS bridges

As previously referred, ICARUS also has various service sets, many inhered from JAUS. As examples, we have the core service set (for vital services, such as transport, events or discovery), mobility service set (for mobile platform services), environment sensing service set (for platform-independent sensor capabilities) and finally manipulator service set (for platform-independent capabilities common across all serial manipulator types). The different services are shown in Figure 5-25. For each of these services sets of messages and protocols for data exchange were defined.

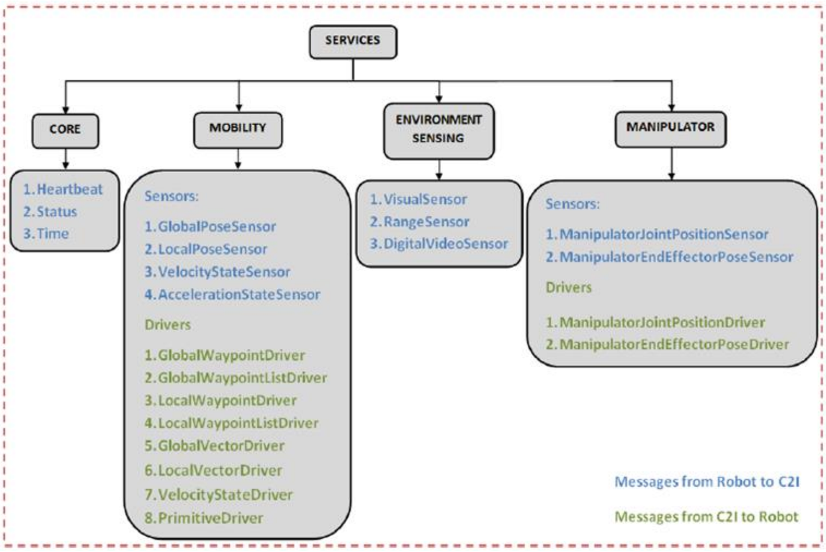

Figure 5-25 - ICARUS Services

The ICARUS JAUS integration can be primarily divided into two separate ICARUS sub-systems: JAUS UxV and JAUS C2I. JAUS UxV integrates the functionalities of the separate vehicles. It establishes the interface between the vehicle and the C2I, providing two separate services, one being the transmission of information collected by the vehicles sensors to the C2I and other the access to actions performed by the vehicle, such as waypoint navigation. The JAUS C2I has the ability of retrieving information from the fleet through the JAUS fleet handler, or of monitoring a single vehicle using the JAUS robot handler[295].

The R2C (or ground segment) is responsible for the vehicle's tasking and control and is housed in a ruggedized box (see Figure 5-26).

Figure 5-26 - R2C box

Is the Command and Control Interface to the system

The UAVs used in the ICARUS project were the following: Atlantik Solar developed by Eidgenoessische Technische Hochschüle Zürich (ETHZ); and the quadrator and Skybotix Indoor Multirotor developed by Associació Catalana d'Empreses constructores de Motlles i Matrius (ASCAMM). These vehicles are presented in Figure 5-27.

Figure 5-27 - ICARUS UAVs

ETHZ Atlantik Solar, ASCAMM Quadrator and Skybotix Indoor Multirotor

The Atlantik Solar is an ETHZ endurance airplane, with a maximum autonomy of 12 hours, and nominal cruise speed of 35 km/h. It integrates a set of sensor systems for SAR missions, and in ICARUS some more sensors were added, using the ICARUS Common Sensing and Processing Unit, that included visual and thermal cameras. This airplane uses ROS/MAVLink, and a JAUS adaptor layer had to be developed for it.

ASCAMM quadrotor is an aerial platform capable of flying autonomously. It follows a pre-planned trajectory, but it can also be teleoperated. It can carry a heavy payload and had already been adapted to search and rescue missions, such as 2D or 3D mapping, but more sensors and a thermal camera were added. This vehicle is also responsible for the survival kit delivery. ASCAMM quadrotor also uses ROS/MAVLink and the same JAUS adaptor was used. The Skybotix Indoor Multirotor is a small vehicle capable of flying both outdoor and indoor. Its primary objective is to enter buildings or areas that are difficult to access in order to search for victims and has obstacle avoidance using his own sensors. It also has a thermal camera to detect victims[296].

Two ground vehicles were used in ICARUS: the Small UGV (SUGV), created by Allen Vanguard and the Large UGV (LUGV), created by Metalliance. They are represented in Figure 5-28. The SUGV is designed to be small and agile, in order to operate in indoor environments, and has a camera and stereo vision system. LUGV is a large track-driven vehicle, which weights approximately four tons. It has laser range finders and stereo vision. It has a heavy gripper and a jackhammer, used to break or grab objects. It is also able to lift the SUGV, using a transport box, so that the SUGV can reach elevated places. These ground vehicles had to be adapted to JAUS so as to fit the ICARUS network[297].

Figure 5-28 - ICARUS UGVs

Large UGV and Small UGV

ICARUS USVs are composed by the Unmanned Rescue Capsule (UCAP), ROAZ II and the U-Ranger. They are represented in Figure 5-29.

Figure 5-29 - ICARUS USVs

ROAZ II, U-Ranger and Rescue Capsule

UCAP and ROAZ II were developed by the Institute for Systems and Computer Engineering, Technology and Science (INESC-TEC). They can be remotely operated from shore or from another vessel, and he can also operate in an autonomous mode, as they are equipped with navigation sensors. These vehicles had to be adapted to create a JAUS Robot entity. The UCAP's purpose is just to transport a life raft over short distances[298]. ROAZ II is a catamaran with electrical propulsion and a a maximum speed of ten knots. It is equipped with a long wave infra-red and a color camera, 3D scanning and a continuous wave RADAR [299]. These sensors can be used to avoid obstacles and for victim detection purposes. This vehicle runs a custom designed control software from its designer, and a JAUS standard was developed for it.

U-Ranger was created by Calzoni and automated by CMRE, initially developed to be used in mine-hunting. However, it is a modular vehicle and it can carry a variety of payloads. It can be controlled manually but it can also follow a plan or execute pre-programmed tasks. This vehicle has a MOOS open architecture. Once again, a JAUS adaptor had to be developed.

In order to validate the interoperability of the system among all these vehicles a number of tests were conducted, first in a laboratory and then in field tests[300]. The laboratory tests were done by means of logged data and simulations to understand if the interface would operate efficiently on a SAR environment. The results showed that all robots had been adapted and integrated into the C2I.

Field tests were conducted in the following places:

- Maritime trials, La Spezia (Italy) 2013;
- Air Trials, Barcelona (Spain) 2014;
- Maritime trials and vehicle cooperation, Lisbon (Portugal) 2014;
- Participation in the EuRathlon Competition 2015;
- Maritime demonstrations simulating a shipwreck, Lisbon (Portugal) 2015;
- Land demonstrations simulating an earthquake, Marche-en-Famenne (Belgium) 2015.

The tests in La Spezia focused on the maritime environment, to test the USV that were being developed to support SAR missions. Several ICARUS partners gathered and tested their vehicles over five days. ROAZ II and the U-Ranger were the vehicles that were tested in this scenario. Although it was the first maritime field trial, they had encouraging results, and they helped to understand what had to be improved in these vehicles.

Multiple air vehicle tests were organized by ASCAMM in the CATUAV Test Center, in Barcelona, in 2014. This is an open rural space, ideal to validate these types of vehicles, and to carry out the studies of integrating multiple aerial vehicles. The two vehicles that were involved in these tests were the Atlantik Solar airplane and the quadrotor. The C2I systems and communications equipment were also validated by its owners in this test. This validation tests were successful, as the quadrotor and the solar airplane could communicate each other and with the C2I.

As for multiple sea vehicle operations, these were performed in Lisbon during the Robotic Exercises (REX) 2014 and 2015 exercises[301]. This is a naval exercise, conducted by the Portuguese Navy and coordinated by CINAV. In the first testes only the ROAZ II and UCAP were used, but many sensors and interoperability issues were tested. In the 2015 tests, dubbed "Lisbon sea trials" a scenario was created to validate project ICARUS premises, simulating a large-scale disaster. In it a ferryboat suffered and accident near the coast. The UAVs swept the area to pinpoint the location of victims and at the same time provided communication relays, allowing for greater range. The USVs would perform rescue missions, that involved approaching the victims and providing floatation and shelter devices. These actions implied testing such capabilities as long endurance flights, vertical takeoff and landing, carrier USVs and UCAPs.

The ICARUS project consortium was also involved in the EuRathlon 2015[302], which was the first multi-domain robotics competition in which teams could demonstrate the intelligence and interoperability of their vehicles in SAR scenarios. This competition was inspired by the 2011 Fukushima accident in which robots should cooperate with each other and search for workers in ruined buildings, leaks of dangerous substances and damage to underwater structures, among other tasks[303]. The maritime part of EuRathlon was focused on underwater robots, which ICARUS does not have, to detect a leak in an underwater pipe, and close the valve that controls it. However, an ICARUS partner, INESC-TEC provided this vehicle. Above water, the objective of this competition is to inspect the inside and outside of a building with UGVs and UAVs, finding entrances and blocked paths to the building, finding a safe path to a machine room inside the building for the UGV, to search for a missing worker in the area, building a map of the different areas, to close the correct valves of leaking pipes and if possible doing this transmitting live position and imagery to the control station. This competition was a success for the ICARUS project. Thanks to the interoperability achieved with all the developments and studies conducted in the project, all the vehicles were able to provide live data during the operations. Therefore, every operator knew where the vehicles were, and the communications were constant and good. The results impressed the organizers, being an evidence of the achieved interoperability. ICARUS project received the Multi-Robot coordination award in this competition.

The C4I capabilities sub-scenario test has the objective of meeting requirements such as deploying ICARUS communication system, providing active links to control stations, establishing a communication network, using JAUS protocol for communication between internal and external subsystems, among other tasks.

The air-air vehicle capabilities test has the objective of meeting requirements such as importing crisis data from control stations, remote control the

UAV, map GPS defined areas, retrieve visual and IR data, search for victims with the UAV, assess victims medical state through the UAV, deploy rescue kits with the UAV, support multiple unmanned SAR missions, among other tasks.

The air-marine vehicle capabilities test has the objective of meeting requirements such as: allow for UCAP autonomous functioning, aiding four victims in the water, retrieve visual data and area mapping, detecting victims, searching for victims using the UAV, among other tasks.

The marine-marine vehicle capabilities test has the objective of meeting requirements such as: ability to deploy the U-RANGER USV from a harbor, to remote control the U-RANGER USV, to function the U-RANGER and the UCAP autonomously, to retrieve visual and IR data, to search for human victims, to deploy UCAPs from the U-RANGER, among other tasks.

The air-marine-marine vehicle capabilities test has the objective of meeting requirements such as: ability to deploy ROAZ II USV from a harbor, to remote control UxS, to aid victims in the water, to retrieve visual and IR data, to map the surrounding area, to search for victims, among other tasks.

The validation process in the sea scenario (Figure 5-30), provided the following results:

	Not Validated	Partially Validated	Validated	Total
C4I Capabilities	5	0	7	12
Air-air vehicle capabilities	2	1	19	22
Air-marine vehicle capabilities	3	0	12	15
Marine-marine vehicle capabilities	15	0	13	28
Air-marine-marine vehicle capabilities	0	0	20	20
Totals	25	1	71	**97**
Percentually %	25,7732	1,030928	73,19588	100

Figure 5-30 – ICARUS results of the validation in sea scenario

The development of the land scenario plays out in an urban area in Belgium which as suffered an earthquake. In this area, the Belgian First Aid and Support Team is activated, with the help of the ICARUS tools. The earthquake causes buildings, bridges and other structures to collapse, resulting in a sudden need for fast intervention for the robotic assets. The UAV deployed were the AROT and the Atlantic Solar unit, which provided area mapping, communication relaying and reconnaissance as well as the delivery of survival kits to stranded victims. The UGVs used were the SUGV and the LUGV, with the purpose of mapping buildings and surrounding areas, transporting payloads, moving debris, defining best exit strategies, identifying victims, breaching obstacles, among other tasks. The communication tools employed were based of the JAUS used by the ICARUS UxVs assets. The validation process was evaluated over the execution of six sub-scenarios: C4I integration, C4I mission planning, deployment, apartments, school and warehouse.

The C4I integration scenario was developed primarily to test the ability of establishing a successful network on the field which allowed to deploy the ICARUS communication system, to connect to an external communication provider, to import map data from Geographic Information System (GIS) provider, to exchange data with the C4I system, among other tasks.

The C4I mission planning scenario tests the assignment of sectors and tasks to SAR teams, by transmitting compiled and integrated information from various data sources. As such, requirements like the ability of importing crisis data, to plan a data-acquisition mission via the C2I, to share mission plans, to remote control the UAVs, to map a GPS-defined zone, to retrieve visual data, to store incoming data, to provide area mapping, among other tasks, were tested.

The Urban Search and Rescue (USAR) deployment scenario had the objective of testing the deployment capabilities and the integration of the communication and C2I system, as well as to test the network and C2I management capabilities applied to dynamic team and resource allocations. The requirements to be met were as follows: ability to select base of operations, ability to retrieve visual and IR data, ability to perform imagery and mapping of the surrounding area, ability of moving the UxV without slowing down the team movement, ability to deploy the UxV and the respective communication links, among other tasks.

The USAR Apartments scenario (i.e. perform SAR on an apartment building) has the objective of testing and assessing the capabilities of the LUGV and the outdoor rotorcraft and its collaborative operation mode, in the ruins of an apartment building. The requirements to be met were as follows: ability to search for human victims with the UAVs, ability to map the area and to pinpoint the victim's locations, ability to assess the medical state of the victims using the UAVs, ability to access field data, ability for the LUGV to route the victim location, to move debris, to cut through obstacles, and to place objects to stabilize structures, among other tasks. The following USAR "school and warehouse" scenarios share the same purpose and tasks (some more directed to the SGV) as the USAR apartments scenario, providing a change of environmental parameters which contribute to the versatility and value of the ICARUS project.

The results obtained, shown in Figure 5-31, prove the success rate of the experiment, since it allowed for the ICARUS project to achieve important pre-established goals, promoting its quality and success.

	Not Validated	Performance below min acceptance level	Performance below goal level	Performance at or above goal level	Breakthrough performance achieved	Total
C4I_Integration		1		1	1	3
C4I_Mission_Planning				5	12	17
Deployment	3	1	5	14	26	49
Apartments	1	1	2	13	8	25
School	2		5	5	12	24
Warehouse	1		5	8	6	20
Totals	7	3	17	46	65	**138**
Percentually	5%	2%	12%	33%	47%	

Figure 5-31 – ICARUS results of the Land Demonstrations

5.2.2. STANAG 4586 - MAVLink Gateway.

In our research projects we have cooperated with the Portuguese UAV company UAVision, that produces several commercially available UAVs. Internally, they use the MAVLink protocol, reviewed in 3.1.15, and very popular amongst hobbyists and leisure UAVs. UAVision would like to start supplying their UAVs to the armed forces, but many of these would like the UAVs to be compliant with STANAG 4586 given that the U.S. DoD uses this protocol for most of their UAVs. In particular, the NATO standard GCS are supposed to control the vehicles using STANAG 4586. We thus agreed with them to develop a STANAG 4586 to MAVLink gateway. The work was done with the help of two MSc students from the Portuguese Naval Academy that did their dissertations on this theme: Midshipman Carapau, and Midshipman Valério.

Given the short time and well-defined but limited objectives of this project, it was decided that the gateway would only deal with the MAVLink commands actually used by UAVision's UAVs, and the STANAG 4586 messages strictly necessary for that translation. Unknown messages are thus simply ignored.

Another important decision was whether the gateway should run on the UAV or on the GCS. Running on the UAV would require more computing power where resources are expensive and scarce but would make the UAV much easier to integrate in a multinational force. On the other hand, integrating the gateway in the GCS, where computing power is cheaper and easier to obtain, would require changes to rather expensive ground stations and the existence of datalinks capable of dealing with MAVLink. The solution was to develop the gateway to run on a Raspberry Pi computer that can be coupled either to the GCS or integrated in the UAV (since it quite light and has reasonable power requirements).

The first part of the work consisted in studying which MAVLink messages were necessary, what format they had, which STANAG 4586 could be used to translate them, and what information was necessary to fill in the data packets.

A simulator was developed to run on a PC and test the conversion routines. All code was written in Python due to its simplicity and portability. That simulator simplified the debugging of the translation routines, that were then used on the Raspberry Pi. In the end we produced:

1) A translation simulator that runs on a standard windows PC with a user-friendly Graphical User Interface (GUI) that allows us to see how different commands are translated from one protocol to another;
2) A set of core routines in Python to translate from one protocol to another;
3) A prototype system, using a Raspberry Pi, connects via one ethernet port to a MAVLink system, and via another ethernet port to a STANAG 4586 system.

Tests were conducted not only to test the correctness of the translation but also, given the tight time constraints of controlling a UAV at a low level, the time delay introduced by the system. On average the overhead of performing the translation was around 1.5 ms, which does not have a major impact on the required applications.

The project was a success because it made it possible to control a UAVision UAV with a STANAG 4586 compliant GCS. However, the set of messages used is quite limited, and an implementation with the full set would be a daunting task.

The main lessons from this work were that translation between protocols can be messy, somethings get lost in translation, but it is possible.

Further information about this work is available in the dissertations of the Midshipmen [304] and in a paper presented in IEEE Oceans[305].

6

Conclusions

In this chapter we explain how the hypothesis about our research question was confirmed in this thesis, we present the main conclusions of the thesis and how it interacted with other research activities. We also give a summary of the publications that came from our work on the thesis and present our view of future work in this area.

6.1. **From a research question to validation of the hypothesis**

The research question was:

Is there a reference model that describes all components and issues concerning unmanned vehicles that are relevant to achieve interoperability of heterogenous groups of such vehicles, and a standard that following that reference model achieves that interoperability?

The answer to this question is *no*. Most models of UxS are implicit in standards, data models, protocols, or frameworks, so in chapter 3 we introduced the concept of IBBs and reviewed the most relevant ones. We showed that none of them had a sufficiently overarching, detailed, and explicit model that could be used universally with clear advantages over the others. We further consolidated this conclusion by analyzing different UxS, in particular those used in research projects where we were involved. The system architectures in those, while having the same general structure (vehicle-datalink-ground station) varied considerably, without any real need to do so. We thus can answer negatively to the first part of the research question: *There IS NOT a reference model that describes all components and issues concerning unmanned vehicles that are relevant to achieve interoperability of heterogenous groups of such vehicles.*

Our hypothesis was:

It is possible to achieve interoperability amongst heterogeneous unmanned vehicles if they all share a common reference model (which we propose) and use one of the existing communication methods to exchange messages.

We *confirmed* this hypothesis during this thesis. We *did* propose a reference model in chapter 4 and named it RAMP. We *proved* that we can apply it in a number of cases, the projects where we were involved, as explained in chapter 5.1. Although not provable in the general case, we believe that RAMP can represent any UxS. We proved that it *is* possible to achieve interoperability, at the command, control and reporting level, with the results shown in project ICARUS, presented in chapter 5.2.1.

6.2. **Integration with other research activities**

In support of the work done for this thesis, the author participated in various research projects and working groups, and was involved in organizing various research activities, such as conferences, meeting, and NATO Lecture Series.

6.2.1. **Research Projects**

The research projects were crucial to gain insight into the problems of defining system architectures, standardization and interoperability. We have already discussed the contribution of the main ones, we present here a summary.

6.2.1.1. **Autoland**

This project was financed by the *Quadro de Referência Estratégica Nacional* (QREN) national program, with 2 partners, to develop a landing system for UAVs aboard ships, and was presented in chapter 5.1.1. The control of the UAV in the landing phase was meant to be the theme for the author's thesis, but it soon became apparent that more important issues arose from the use of UxS in the Navy, and this project acted as an introduction to the problems of defining system architectures that support interoperability, and thus acted as motivation to start working on RAMP.

6.2.1.2. Seagull

This project was financed by the QREN national program, with 5 partners, to develop a UxS that could be used to detect oil spills and suspicious behaviours of ships, and pass that information to maritime coordination centers, and was presented in chapter 5.1.2. It contributed to this thesis by allowing the author to get further involved in designing and integrating UxS, and due to the issues, that turned up during the project allows the RAMP model to grow and mature.

6.2.1.3. GammaEX

This project was financed by the Portuguese Ministry of Defense, with 5 partners, to develop a UAV (multicopter) to be used in dangerous environments to detect chemical and radiological agents and was presented in chapter 5.1.3. This project was the first with a distinctly military nature, but with an emphasis on complying with international standards such as ATEX for safe operation in explosive environments. Since the author was involved from the onset in the definition or requisites, the system architecture of the UxS used follows RAMP completely, and thus the project was useful to consolidate its usefulness.

6.2.1.4. ICARUS

This project was financed by the EU FP7 program, with 24 partners from 10 countries, to develop robotic tools for search-and-rescue operations and was presented in chapter 5.2.1. It was the first large international program in which we were involved, which by itself was an enriching experience. The main contribution to this thesis was needing to have various UxS with very different characteristics and very different origins cooperating with each other. Not only was RAMP a common way of describing the various vehicles, but more importantly it was possible to agree on a standard, JAUS, to command and control all the vehicles from a single ground station.

6.2.1.5. SUNNY

This project was financed by the EU FP7 program, with 18 partners from 11 countries, to develop a 2-tier system of UAV to detect illegal immigration and other illegal activities in the seas around the European Union[306],[307]. This project (Figure 6-1), although taking up time, was less relevant to this thesis because the means used to achieve the required interoperability relied only in communication amongst GCS[308]. The difficulties encountered are, by themselves, and eloquent defense of our thesis: is it better to agree to common IBBs than to simply exchange high-level information. The approach followed in SUNNY is similar to that followed by many international (an even national) research projects because, at first sight, it is easier and faster to let each partner continue using their own tools and standards than to force everyone to change to a common one. Moreover, there seldom is agreement about which IBB to use, and in the discussions, there is a lot of misunderstanding due to the absence of an agreed model, syntax and semantics. Even issues related to terms such as RPAS, Drone, UAV, AUV, UUV, *etc.* can be terribly confusing.

Figure 6-1 – SUNNY Final Tests

Photographed during the final demonstration of research project SUNNY in April 2018, São Jacinto, Portugal

6.2.2. Working Groups

6.2.2.1. GT – VENT (Portuguese Navy working group for unmanned systems)

GT-VENT is a Portuguese Navy working group, stet up in 2013 by the Navy's General Staff, to implement the capability of using UxS in the Navy. In fact, the Navy has been using UUV operationally since 2009, but that capability was achieved an ad-hac approach. Within NATO, when forces need a given capability they should follow the *Joint Capabilities Integration Development System*, JCIDS, most commonly known as DOTMLPFI[309]. The initials stand for Doctrine, organization, training, materiel, leadership and education, personnel, facilities, and interoperability. GT-VENT must thus address all these issues related to UxS. The first steps consisted of defining a roadmap and Doctrine for the use of UxS. This requires planning *how, when and by whom* the different elements will fall into place and defining CONOPS for this new capability. The author has been a member of this working group since its inception and is still continuing with his expertise to the various tasks that have to be performed. The work done in this Task Group contributed to this thesis by giving a broader view of administrative and organizational issues of the use of UxS and their importance for the armed forces.

6.2.2.2. SCI-ET-009 Command and Reporting Standards and Associated Development Tools for UxS

NATO's Science and Technology Organization is based in Brussels and reports directly to the Military and Political Council of NATO. Within this organization, there is an office, named Collaboration Support Office (CSO) that promotes cooperative research amongst NATO nations. It is the follow-on of AGARD, created in the 1952 by Doctor Theodore Von Karman to help NATO achieve scientific and technological superiority. It later changed names, to RTA, RTO, cooperated in setting up the ASI program, summer schools, lecture series, scholarships, etc. Globally all these initiatives had a considerable impact in science and technology throughout NATO.

Presently, the CSO has 7 panels, divided by themes, where scientists from NATO nations get together (each financed by his own nation) to work on different themes. The *Systems, Concepts, and Integration* panel (SCI) deals with systems of systems, systems that require a tight integration of interdisciplinary areas, and other issues[310]. Within each panel there are Exploratory Teams (ET -1 year initiative of at least 4 nations to explore a theme and decide if further work is necessary), Research Task Groups (RTG – 3 to 4 year task groups with at least 4 nations to study a given theme), Lecture Series (LS – short 2 day courses, given in at least 3 nations), Symposia (SY-medium to large scientific conferences), Specialist Meeting (SM- Small conferences, usually by invitation), and others.

The SCI-ET-009 was an Exploratory team setup in 2014 to study *Command and Reporting Standards and Associated Development Tools for UxS*. The countries involved were: Belgium, Spain, Portugal, U.S. and CMRE (that as Cooperating Organization has the same status as nations). This team had meetings at the Portuguese Naval Academy in Lisbon and at the Royal Military Academy in Brussels. The result of the work and discussions amongst partners resulted in the planning of a Lecture Series (LS) on "Command and Reporting Standards and

Development Tools for UxS”. This ET contributed to this thesis by strengthening the need for standards and giving the author a better understanding of STANAG 4586, JAUS, MOOS, and MAVLink.

6.2.2.3. **Lecture Series SCI – 271 Command and Reporting Standards and Associated Development Tools for UxS**

NATO Lecture Series are short courses, usually with a duration of 2 days, that are given in at least three different locations, by experts in the field, and funded by NATO itself. Lecture notes are produced and made available to all NATO nations at the CSO website, and an independent evaluator is sent to at least one location to write a report on the course.

SCI – 271 Command and Reporting Standards and Associated Development Tools for UxS was the result of the work done and recommendations given by SCI-ET-009 (see above). The syllabus of the Lecture Series was:

- Review the need to have standards to facilitate coordination and cooperation of unmanned systems.
- Overview existing interoperability standards;
- Overview some open source tools that can be used in the development of standard compliant unmanned vehicles;
- Overview, amongst others the following standards and frameworks: STANAG 4586, JAUS, MOOS and ROS;
- Hands on tutorials.

The lectures were given in Lisbon (PRT) 26-27 January 2015, La Spezia (ITA) 09-10 February 2015 and in Brussels (BEL) 12-13 February 2015. The chairman of the lecture series was Daniel Serrano, from the Spanish company ASCAMM, and the lectures series were given by Daniel Serrano, Alberto Gratti, from CMRE, and the author.

The lectures were very successful and had considerable attendance by students, university professors, military personnel that work with UxS, researchers and engineers from companies and research centers.

This lecture series contributed to this thesis by establishing a board base of support for the ideas proposed in this thesis and by forcing the author to study in depth the issues addressed, with emphasis on STANAG 4586.

6.2.2.4. SCI-ET-012 Affordable Robotics for Military Operations

The SCI-ET-009 was an Exploratory team setup in 2014 to study *Affordable Robotics for Military Operations*. The countries involved were: Belgium, Estonia, France, Germany, Portugal, U.S. and CMRE (that as Cooperating Organization has the same status as nations). This team had a physical meeting at the Portuguese Naval Academy in Lisbon and several teleconferences.

The primary objectives of the ET were to develop an understanding of the major cost drivers in the design and deployment of robotic systems. To get this information a survey was conducted amongst the major stakeholders: industry, procurement and acquisition staff from the armed forces and DoD, researchers, academics, and operational staff.

One of the major enthusiasts of this ET was later Chief Scientist of NATO (Doctor Thomas Killion) that defended that some UxS should bypass traditional military acquisition processes and be designed for *short life-cycle*. As a consequence, the follow-on to this ET focused on that aspect, and the author did not participate in those initiatives.

The main contribution to this thesis was the confirmation that industry-wide standards would in fact help decrease the cost of military UxS.

6.2.2.5. STANAG 4586: Standard Interfaces of UAV control systems (UCS) for NATO UAV interoperability

STANAG 4586 was reviewed in detail in 3.1.1. The objective of this group is to maintain and update STANAG 4586 so as to provide a document which is technically correct, supports the user requirements/capabilities and incorporates the most recent technologies providing efficient operation and control of UAS.

As with all STANAGs, there is a designated custodian of the standard, in this John Mayer from the United States Office of Naval Research, that convenes the contributors to the STANAG when necessary. In the case of STANAG 4586, given the fast pace of technological changes and the importance of UAVs, there have been two meetings every year. The NATO nations currently involved in the update of the standard are: Albania, Belgium, Bulgaria, Canada, Croatia, Czech Republic, Denmark, Estonia, France, Germany, Greece, Hungary, Iceland, Italy, Latvia, Lithuania, Luxembourg, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Turkey, UK, US, Norway, Montenegro and Israel.

The author participated by email and teleconference in the work done to propose a revision of STANAG 4586, and organized the meeting of this group in Lisbon, during which he was for the first time physically present at the meeting (in 23-26 May 2016)

The participation in this group contributed to this thesis by giving a better insight to the problems of negotiating and approving international standards and understanding better the problems with STANAG 4586 and the issues with its various versions.

6.2.2.6. **NIAG SG 202 - Study on development of conceptual data model for a multi - domain unmanned platform control system**

NATO Industrial Advisory Group (NIAG) is a NATO organization, under the Council of NATO National Armaments Directors (CNAD) where industries from NATO nations can help the armed forces in developing, choosing, and operating military equipment. NIAG establishes *Study Groups* to address specific issues where industrial partners from NATO nations can participate and give their view on what should be done. While NIAG groups are aimed mainly at industry, governmental representatives are welcome and act as non-voting advisors and experts (mainly on the operational use of the systems).

Because of the involvement in STANAG 4586, the author participated in NIAG *SG 202 - Study on development of conceptual data model for a multi-domain unmanned platform control system*, having attended two meetings (in Lisbon and in Italy). This study group was setup in 2015 and finished in 2016. It had participants from: Albania, Belgium, Bulgaria, Canada, Croatia, Czech Republic, Denmark, Estonia, France, Germany, Greece, Hungary, Iceland, Italy, Latvia, Lithuania, Luxembourg, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Turkey, UK, and U.S.

The standard that was produced by this study group and its predecessor NIAG 157 was presented in chapter 3.1.121. According to the group's documentation, "*The aim of this Study Group is to develop a data model that would represent all the information required for a Control System to operate assets from multiple domains, and to develop draft guidance on how to implement and test the system. A secondary objective is to propose a plan for NATO development of a prototype*"

Working with this study group had an impact on this thesis mainly do fine-tune the concepts related to the ground segment of RAMP.

6.2.2.7. **SCI - RTG – 288 on Autonomy in Limited Communications Environments**

The SCI - RTG - 288 is a CSO task group setup in 2015, and originally scheduled to end in 2018, to address the problem of *Autonomy in Limited Communications Environments*. When the RTG started, the main concern was with the low bandwidth available in underwater communications for controlling UUVs. With time, is soon became clear that the RTG needed to have common standards for controlling those UUVs, and we joined the RTG in 2017 to work on that issue. The countries currently involved are: Canada, France, UK, Italy, U.S., Nederland, Norway, Turkey, Portugal, and CMRE. The author participated in one of the meetings in Paris (in 2017) and has participated in the work by email and teleconference.

The work in this RTG contributed to this thesis by making clear the need for universally accepted standards, but also to raise awareness for the issues related to simplifying the messages in special cases, such as very low bandwidth acoustic communications.

6.2.3. **Member of Organizing Committees**

As a corollary to the work on this thesis the author was a member of the organizing committees of various science related events. The participation in these committees contributed to this thesis by proving a networking environment that enhances opportunities to exchange ideas and discuss the themes of this thesis with experts from around the world. It is also a way of learning "the ropes of the trade" of research and giving back to the community. The author was a member of the organizing committees of the following events:

- Robotic Exercise (REX'14) 2014, 30 June to 04 July 2014, Lisbon;
- Lecture Series SCI – 271 Command and Reporting Standards and Associated Development Tools for UxS, 26 to 27 January 2015, Lisbon;
- 8th workshop IARP RISE' 2015 – International Advanced Robotics for Risky Environment and Environmental Surveillance, 28 to 29 January 2015, Lisbon;
- 6th Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS'15), 13 to 15 April 2015, Monte Caparica;
- Meeting of NIAG 202 - Study on development of conceptual data model for a multi - domain unmanned platform control system, 20 to 22 April 2015, Lisbon;
- Robotic Exercise (REX'15) 2015, 29 June to 10 July 2015, Lisbon;
- Meeting of STANAG 4586 - Standard Interfaces of UAV control systems (UCS) for NATO UAV interoperability, 23 to 26 May 2016, Lisbon;
- Robotic Exercise (REX'16) 2016, 27 June to 08 July 2016, Lisbon;
- Lecture Series AVT – 274 on Unmanned Air Vehicles: Technological Challenges, Concepts of Operations and Regulatory Issues, 23 to 24 May 2017, Lisbon;
- Robotic Exercise (REX'17) 2017, 11 to 14 July 2017, Lisbon.

6.3. **Publications Summary**

As part of the work done for this thesis, the author published various scientific papers listed below.

6.3.1. **Book Chapters**

- Daniel Serrano, German Moreno, José Cordero, José Sanches, Shashank Govindaraj, Mário Monteiro Marques, Victor Lobo, Sephano Fioravanti, Alberto Grati, Konrad Rudin, Massimo Tosa, Aníbal Matos, André Dias, Alfredo Martins, Janusz Bedkowski, Haris Balta, Geert de Cubber, "Interoperability in a Hereogeneous team of Search and Rescue Robots", in Cubber, Geert De, Daniela Doroftei, Konrad Rudin, Karsten Berns, Daniel Serrano, Jose Sanchez, Shashank Govindaraj, Janusz Bedkowski, and Rui Roda. "Search and Rescue Robotics-From Theory to Practice.", 2017;

- Geert de Cubber, Daniela Doroftei, Haris Balta, Aníbal Matos, Eduardo Silva, Daniel Serrano, Shashank Govindaraj, Rui Roda, Victor Lobo, Mário Monteiro Marques and Rene Wagemans, “Operational Validation of Search and Rescue Robots”, in Cubber, Geert De, Daniela Doroftei, Konrad Rudin, Karsten Berns, Daniel Serrano, Jose Sanchez, Shashank Govindaraj, Janusz Bedkowski, and Rui Roda. "Search and Rescue Robotics-From Theory to Practice.", 2017.

6.3.2. **Papers published in journals**

Mario Monteiro Marques, V. Lobo, R. Batista, J. Oliveira, A. P. Aguiar, J. E. Silva, J. B. de Sousa, M. de F. Nunes, R. A. Ribeiro, A. Bernardino, and J. S. Marques, “An unmanned aircraft system for maritime operations,” International Journal of Advanced Robotic Systems, vol. 15, no. 4, p. 172988141878633, 2018.

6.3.3. **Papers presented in conferences**

- Júlio Carvalho, Wilson Antunes, Tiago Goncalves, Mario Monteiro Marques, Victor Lobo “Unmanned aerial vehicles in chemical, biological and nuclear environment: sensors review and concept of operations”, 13 IARP Workshop on Humanitarian Demining and Similar Risky Interventions HUDEM 2015, Croacia, 2015, pp.1-4;
- Mario Monteiro Marques, P. Dias, N. Santos, V. Lobo, R. Batista, D. Salgueiro, R. Ribeiro, J. Marques, A. Bernardino, M. Griné, M. Taiana, M. Nunes, E. Pereira, J. Morgado, A. Aguiar, M. Costa, J. Silva, A. Ferreira, J. Sousa, “Unmanned Aircraft Systems in Maritime Operations: Challenges addressed in the scope of the SEAGULL project,” in MTS/IEEE OCEANS 2015, Génova, 2015, pp. 1–6;
- Filipe Morais, Tiago Ramalho, Pedro Sinogas, Mario Monteiro Marques, Nuno Santos, Victor Lobo, “Trajectory and Guidance Mode for autonomously landing an UAV on a naval platform using a vision approach,” in MTS/IEEE OCEANS 2015, Génova, 2015, pp. 1–7;

- Mario Monteiro Marques, Gonçalo Rosa, Fernando Coito, Victor Lobo "Two Major Architectures for Unmanned Systems – STANAG 4586 and JAUS,", International Conference on Informatics, Control and Automation, Phuket, 2015, pp. 1–6;
- Mario Monteiro Marques, Alfredo Martins, Anibal Matos, Nuno Cruz, José Miguel Almeida, José Carlos Alves, Victor Lobo, Eduardo Silva, "REX14 – Robotic Exercises 2014 – Multi-robot field trials," in MTS/IEEE OCEANS 2015, Washington, 2015, pp. 1–6;
- Mario Monteiro Marques, Rui Parreira, Victor Lobo, Alfredo Martins, Aníbal Matos,Nuno Cruz, José Miguel Almeida, José Carlos Alves, Eduardo Silva, Janusz Będkowski, Karol Majek, Michał Pełka, Paweł Musialik, Hugo Ferreira, André Dias,Bruno Ferreira, Guilherme Amaral, André Figueiredo, Rui Almeida, Filipe Silva, Daniel Serrano, German Moreno, Geert De Cubber, Haris Balta, Halil Beglerović, Shashank Govindaraj, José Manuel Sanchez, Massimo Tosa, "Use of multi-domain robots in search and rescue operations – contributions of the ICARUS team to the euRathlon 2015 challenge," in MTS/IEEE OCEANS 2016, Xangai, 2016, pp. 1–7;
- Miguel Duarte, Jorge Gomes, Vasco Costa, Tiago Rodrigues, Fernando Silva,Vıctor Lobo, Mario Monteiro Marques, Sancho Moura Oliveira, and Anders Lyhne Christensen, "Application of Swarm Robotics Systems to Marine Environmental Monitoring," in MTS/IEEE OCEANS 2016, Xangai, 2016, pp. 1–8;
- Mario Monteiro Marques, Júlio Gouveia-Carvalho, Ricardo Pascoal, Cristina Matos, "ATEX legal and standard framework applied to UAS in Mine Action and other risky interventions," 14 IARP Workshop on Humanitarian Demining and Similar Risky Interventions HUDEM 2016, Croacia, 2016, pp.1-4;
- Pedro Castro Fernandes, Mario Monteiro Marques, Victor Lobo, "Barlavento – Considerations about the Design of an Autonomous Sailboat," World Robotics Sail Conference, Viana do Castelo, 2016, pp.1-14;

- Mario Monteiro Marques, Victor Lobo, Ricardo Batista, J. Almeida, Ricardo Ribeiro, Alexandre Bernardino, Maria de Fátima Nunes, "Oil Spills Detection: Challenges addressed in the scope of the SEA-GULL project," in MTS/IEEE OCEANS 2016, Monterey, 2016, pp. 1–6;
- Ricardo Mendonça, Mario Monteiro Marques, Francisco Marques, André Lourenco, Eduardo Pinto, Pedro Santana, Fernando Coito, Victor Lobo and José Barata, "A Cooperative Multi-Robot Team for the Surveillance of Shipwreck Survivors at Sea," in MTS/IEEE OCEANS 2016, Monterey, 2016, pp. 1–6;
- Mario Monteiro Marques, V. Lobo, Júlio Gouveia-Carvalho, Alfredo José Martins Nogueira Baptista, Jorge Almeida, Cristina Matos, Rodolfo Santos Carapau, Alexandre Valério Rodrigues, "CBRN remote sensing using Unmanned Aerial Vehicles: Challenges addressed in the scope of the GammaEx project regarding hazardous materials and environments," in 6th International Conference on Risk Analysis and Crisis Response (RACR-2017), Czech Republic, 2017, pp. 1–6;
- Mario Monteiro Marques, Augusto Salgado, Victor Lobo, Rodolfo Santos Carapau, Alexandre Valerio Rodrigues, Marc Carreras, Joseta Roca, Narcís Palomeras, Natàlia Hurtós, Carles Candela, Alfredo Martins, Aníbal Matos, Bruno Ferreira, Carlos Almeida, Filipe Aranda de Sa, José Miguel Almeida, Eduardo Silva, "STRONGMAR Summer School 2016 – Joining theory with a practical application in Underwater Archeology," in MTS/IEEE OCEANS 2017, Alberdeen, 2017, pp. 1–6;
- Rodolfo Santos Carapau, Alexandre Valério Rodrigues, Mario Monteiro Marques, Victor Lobo, Fernando Coito, "Interoperability of Unmanned Systems in Military Maritime Operations: Developing a controller for unmanned aerial systems operating in maritime environments," in MTS/IEEE OCEANS 2017, Alberdeen, 2017, pp. 1–7;

- Alexandre Valério Rodrigues, Rodolfo Santos Carapau, Mario Monteiro Marques, Victor Lobo, Fernando Coito, “Unmanned Systems Interoperability in Military Maritime Operations: MAVLink to STANAG 4586 Bridge,” in MTS/IEEE OCEANS 2017, Alberdeen, 2017, pp. 1–5;
- Mario Monteiro Marques, Rodolfo Santos Carapau, Alexandre Valério Rodrigues, V. Lobo, Júlio Gouveia-Carvalho, Wilson Antunes, Tiago Gonçalves, Filipe Duarte, Bernardino Verissimo, “GammaEx project: A solution for CBRN remote sensing using Unmanned Aerial Vehicles in maritime environments,” in MTS/IEEE OCEANS 2017, Anchorage, 2017, pp. 1–6;
- Mario Monteiro Marques, Mario Gatta, Miguel Barreto, V. Lobo, Aníbal Matos, Bruno Ferreira, Paulo J. Santos, Paulo Felisberto, Sérgio Jesus, Frederich Zabel, Ricardo Mendonça, Francisco Marques, “Assessment of a shallow water area in the Tagus estuary using Unmanned Underwater Vehicle (or AUV's), vector-sensors, Unmanned Surface Vehicles, and Hexacopters – REX’17,” in MTS/IEEE OCEANS 2018, Kobe, 2018, pp. 1–5.

6.3.4. **Posters presented in conferences**

Mario Monteiro Marques, Victor Lobo and Fernando Coito “Reference Model for Interoperability of Autonomous Systems” 6thDoctoral Conference on Computing, Electrical and Industrial Systems 2015, Caparica, Portugal

6.3.5. **Invited Oral Presentations**

- SEACON Project – Undersea Robotics Supporting Navy Operations, ICT2014, Lisbon, 07 May 2014;
- SEAGULL - Intelligent Systems to support maritime awareness based on Unmanned Aerial Vehicles, 3rd Workshop on European Unmanned Maritime Systems, Oporto, 30 May 2014;
- Drones e veículos autónomos: desafios do presente e do futuro, 8º Congresso do Comité Português da URSI, Lisbon, 28 November 2014;

- RPAS could bring to the search and rescue activities, ESA – EMSA Workshop "Remotely Piloted Aircraft Systems for maritime surveillance, Lisbon, 28 e 29 October 2015.

6.4. **Future Work**

There are several issues related to the work done for this thesis that require future work and that can be very relevant for this area.

The most important issue is to consolidate RAMP, which can be done at various levels:

Formal Approval as a Standard – A reference model such as RAMP may be very useful to structure ideas and describe a vehicle, but its usefulness is directly proportional to the support it gets throughout the community. That support depends naturally on its intrinsic value, but science and technology history are littered with great ideas that were wasted because few people knew about them. Writing about RAMP, publishing it in journals and presenting it at conferences may be useful, but it probably not the best road to success. A paper on RAMP would be hard to publish on a top journal and would not provide good reading because it is, in essence, just a list. A better way to make it know would be to get it approved as an international standard. This would expose it to a wide audience, and if suppliers to large buyers (mainly military forces) were required to describe their systems with RAMP, it would be studied with greater detail, and consequently used much more. Thus, we feel that the effort to have RAMP approved in NATO as a STANAG or in other standardization organizations would be very useful.

Detail and Coverage - We believe that what we already produced is useful but it is by no means complete. At the third hierarchical level (the Sub-Systems) there may be room to define more categories, so as to avoid overloading the generic "others" with sub-systems that may be common to many vehicles. The forth level of the hierarchy is even more open to improvement, although being so specific it would only make sense if other improvements (CAE tools, functional relations, and others that we shall see later) were also available.

Support Software - This reference model is not complicated, but its use and understanding would benefit from Computer Aided Engineering (CAE) tools. These tools can be used to store information about the different components, to classify them in the right category, to detect overlaps, guarantee redundancies, and compute various parameters. For example, if we had libraries with the characteristics of the different components (with data about their weight, size, capabilities, interfaces, etc), we might be able to do fast prototyping and design of UxS by trying out different configurations. If the vendors provided information about their systems in a machine-readable format, this would be event simpler. This software would also allow easier real-time interoperability by providing a means of one UxV declaring its capabilities to a system, allowing "plug & play" integration of UxVs in multi-vehicle UxS.

Formal Functional Description – The existing RAMP defines what the components are, but not exactly what they do or how. A more complete model would formally define the functions of each component, making interfacing much simpler. As an example, the MB1.MS4.SS1 (image sensor) could have a formal definition of capabilities, including types of commands and types of information provided. This sill falls short of a complete protocol definition but shortens the gap and makes choosing a protocol much simpler. Even without a complete protocol, this formal functional description would enable the development of conceptual simulators to test the feasibility of UxS for given tasks.

Mapping Existing IBBs to RAMP – In this thesis we reviewed various standards, protocols, data models, frameworks, and reference architectures, which we generically called IBBs, and when possible showed their relation to RAMP. We did not, however, map them formally to RAMP, or "populate" RAMP with the existing IBBs. This is a necessary task when we need to choose which IBBs to use on a given system.

Educational Tools – As previously stated, science and technology are only useful if they are known and used. During this thesis we tried to promote education in this area, through NATO Lecture Series and their lecture notes, and our own classes and notes, but more educational material is certainly needed. It would be important to have a website with reference material (formal descriptions), tutorials, supporting software, reference to published papers, and other teaching materials.

Bibliography

[1] G. Cai, B. M. Chen, and T. H. Lee, "An overview on development of miniature unmanned rotorcraft systems," *Frontiers of Electrical and Electronic Engineering in China*, vol. 5, no. 1. pp. 1–14, 2010.

[2] C. E. Nehme, S. D. Scott, M. L. Cummings, and C. Y. Furusho, "Generating Requirements for Futuristic Hetrogenous Unmanned Systems," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 50, no. 3, pp. 235–239, 2006.

[3] M. Degarmo and G. M. Nelson, "Prospective Unmanned Aerial Vehicle Operations in the Future National Airspace System," *System*, no. September, pp. 1–8, 2004.

[4] R. Mendonc, M. M. Marques, F. Marques, E. Pinto, P. Santana, F. Coito, and V. Lobo, "A Cooperative Multi-Robot Team for the Surveillance of Shipwreck Survivors at Sea," in *Proceedings of IEEE/MTS OCEANS 2016, Monterey*, 2016, pp. 1–6.

[5] S. Rowe and C. R. Wagner, "An introduction to the joint architecture for unmanned systems (JAUS)," *Ann Arbor*, vol. 1001, p. 48108, 2008.

[6] R. Madhavan, R. Lakaemper, and T. Kalmar-Nagy, "Benchmarking and standardization of intelligent robotic systems," *Adv. Robot. Int. Conf.*, pp. 1–7, 2009.

[7] P. Taylor, D. W. Casbeer, D. B. Kingston, R. W. Beard, T. W. Mclain, D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. M. C. Lain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles Cooperative forest fire surveillance using a team of small unmanned air vehicles," no. September 2012, pp. 37–41, 2011.

[8] P. Mac and E. Organick, *The Multics System: An Examination of Its Structure*. 1972.

[9] W. Babich, R. Simpson, R. Thall, and L. Weissman, "The Ada Language System," vol. 75, no. June, pp. 37–45, 1981.

[10] S. Crawford and L. Stucki, “Peer review and the changing research record,” *J. Am. Soc. Inf. Sci.*, vol. 41, no. 3, pp. 223–228, 1990.

[11] L. Camarinha-Matos, *Sientific Resarch Methodologies and Techniques - Unit 2: Scientific Method*. 2014.

[12] K. A. John F. Hughes, Andries van Dam , Morgan McGuire , David F. Sklar, James D. Foley , Steven K. Feiner, *Computer Graphics: Principles and Practice (3rd Edition)*, 3 edition. Addison-Wesley Professional, 2013.

[13] A. P. S. Mohammad Jamshidi, *System of Systems Engineering: Innovations for the 21st Century 1st Edition*, 1 edition. Wiley, 2008.

[14] R. S. Carapau, A. V. Rodrigues, M. M. Marques, V. Lobo, and F. Coito, “Interoperability of unmanned systems in military maritime operations: Developing a controller for unmanned aerial systems operating in maritime environments,” in *Proceedings of IEEE/MTS OCEANS 2017, Alberdeen*, 2017, vol. 2017–Octob, pp. 1–7.

[15] A. Geraci, F. Katki, L. McMonegal, B. Meyer, and H. Porteous, “IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries,” *IEEE Std 610*. p. 1, 1991.

[16] “Information Technology Standards and Organizations Glossary.” [Online]. Available: https://whatis.techtarget.com/definition/framework. [Accessed: 16-Apr-2018].

[17] M. Broy, M. V. Cengarle, H. Grönniger, and B. Rumpe, “Definition of the System Model,” *UML 2 Semant. Appl.*, pp. 61–93, 2009.

[18] N. Group, “NIAG Study on Multi-Domain Unmanned Vehicle Control (SG.157),” 2012.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an opensource Robot Operating System,” *ICRA Work. open source Softw.*, vol. 3, p. 5, 2009.

[20] HIMSS, “What is Interoperability,” *HIMSS Dictionary of Healthcare Information Technology Terms*. [Online]. Available: http://www.himss.org/library/interoperability-standards/what-is-interoperability. [Accessed: 03-Sep-2016].

[21] Department of Defense, “DoD Satellite Communications (SATCOM),” 2016.

[22] NATO, “Interoperability for joint operations,” no. July, p. 9, 2006.

[23] D. J. W. A. S. Tanenbaum, *Computer Networks*, 5th editio. Pearson Higher Education, 2013.

[24] A. L. Russell, “‘Rough consensus and running code’ and the Internet-OSI

standards war," *IEEE Ann. Hist. Comput.*, vol. 28, no. 3, pp. 48–61, 2006.

[25] J. Iden and T. R. Eikebrokk, "The impact of senior management involvement, organisational commitment and group efficacy on ITIL implementation benefits," *Inf. Syst. E-bus. Manag.*, vol. 13, no. 3, pp. 527–552, 2015.

[26] S. E. McIntosh, "The Wingman-Philosopher of MiG Alley: John Boyd and the OODA Loop.," *Air Power Hist.*, vol. 58, no. 4, pp. 24–33, 2011.

[27] B. Brehmer, "The Dynamic OODA Loop: Amalgamating Boyd's OODA Loop and the Cybernetic Approach to Command and Control," *Proc. 10th Int. Command Control Res. Technol. Symp. Futur. C2*, no. December, 2005.

[28] G. R. Hasegawa, *Villainous Compounds: Chemical Weapons and the American Civil War*. SIU Press, 2015.

[29] "Remote Piloted Aerial Vehicles : An Anthology." [Online]. Available: http://www.ctie.monash.edu.au/hargrave/rpav_home.html. [Accessed: 12-Mar-2018].

[30] M. E. Peterson, "The UAV and the current and future regulatory construct for integration into the national airspace system," *J. Air L. Com.*, vol. 71, p. 521, 2006.

[31] "The 'Aerial Target' and 'Aerial Torpedo' in Britain." [Online]. Available: http://www.ctie.monash.edu.au/hargrave/rpav_britain.html. [Accessed: 12-Mar-2018].

[32] "Hewitt-Sperry Automatic Airplane." [Online]. Available: https://en.wikipedia.org/wiki/Hewitt-Sperry_Automatic_Airplane. [Accessed: 12-Mar-2018].

[33] J. F. Keane and S. S. Carr, "A Brief History of Early Unmanned Aircraft," *John Hopkins APL Tech. Dig.*, vol. 32, no. 3, pp. 558–571, 2013.

[34] "Fritz X." [Online]. Available: https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Fritz_X.html. [Accessed: 12-Mar-2018].

[35] W. Wolf, *German Guided Missiles: Henschel HS 293 and Ruhrstahl SD 1400X*, vol. 53. Merriam Press, 1997.

[36] "V-1 the German Flying Bomb." [Online]. Available: https://www.bundesarchiv.de/EN/Navigation/Meta/Kontakt/kontakt.html. [Accessed: 12-Mar-2018].

[37] J. D. Blom, *Unmanned Aerial Systems: a historical perspective*. 2010.

[38] "Operation Aphrodite." [Online]. Available: https://en.wikipedia.org/wiki/Operation_Aphrodite. [Accessed: 12-Mar-2018].

[39] "Ryan Firebee." [Online]. Available: https://en.wikipedia.org/wiki/Ryan_Firebee. [Accessed: 12-Mar-2018].

[40] "Helicopter Historical Foundation." [Online]. Available: http://www.gyrodynehelicopters.com/. [Accessed: 12-Mar-2018].

[41] "Naval Drones." [Online]. Available: http://www.navaldrones.com/pioneer.html. [Accessed: 12-Mar-2018].

[42] "Another US Predator Drone Crashes in Turkey." [Online]. Available: https://sputniknews.com/world/201708231056735486-us-predator-drone-crash-turkey. [Accessed: 12-Mar-2018].

[43] "ScanEagle." [Online]. Available: https://www.boeing.com/history/products/scaneagle-unmanned-aerial-vehicle.page. [Accessed: 12-Mar-2018].

[44] J. L. Coggeshall, "The Fireship and its Role in the Navy." p. 235, 1997.

[45] T. Czapla and J. Wrona, *Technology development of military applications of unmanned ground vehicles*. 2013.

[46] R. Branfill-Cook, *Torpedo: The Complete History of the World's Most Revolutionary Naval Weapon*. Naval Institute Press, 2014.

[47] H. R. Everett and M. Toscano, *Unmanned systems of World Wars I and II*. MIT Press, 2015.

[48] M. M. Graham, "UNMANNED SURFACE VEHICLES : An Operational Commander's Tool for Maritime Security," vol. 298, no. 704, pp. 0–35, 2008.

[49] N. Friedman, *The Naval Institute Guide to World Naval Weapons Systems*. Naval Institute Press, 1994.

[50] J. Manley, "Unmanned surface vehicles, 15 years of development," *Ocean. 2008*, pp. 1–4, 2008.

[51] "Autonomous Surface Craft Development." [Online]. Available: http://web.mit.edu/jmanley/www/Projects.html. [Accessed: 12-Mar-2018].

[52] P. C. Fernandes, M. M. Marques, and V. Lobo, "Barlavento-considerations about the design of an autonomous sailboat," in *World Robotics Sail Conference 2016*, 2016, pp. 19–30.

[53] "Tale of the Teletank: The Brief Rise and Long Fall of Russia's Military Robots." [Online]. Available: https://www.popsci.com/blog-network/zero-moment/tale-teletank-brief-rise-and-long-fall-russia's-military-robots. [Accessed: 12-Mar-2018].

[54] D. Michel, A. H., & Gettinger, "Out of the Shadows: The Strange World of Ground Drones," 2013.

[55] T. H. E. National and A. Press, *Technology Development for Army Unmanned*

Ground Vehicles. 2003.

[56] P. F. Lt and K. X. Lt, "KERVEROS I : An Unmanned Ground Vehicle for Remote-Controlled Surveillance," vol. 4, no. 1, pp. 223–236, 2014.

[57] "Stanford Cart How a Moon Rover Project was Blocked by a Politician but got Kicked by Football into a Self-Driving Vehicle." [Online]. Available: https://web.stanford.edu/~learnest/les/cart.html. [Accessed: 12-Mar-2018].

[58] S. Odedra, S. Prior, and M. Karamanoglu, "Investigating the Mobility of Unmanned Ground Vehicles," *Int. Conf. Manuf. Eng. Syst. Proc.*, no. January, pp. 380–385, 2009.

[59] "Schappell Automation Corporation." [Online]. Available: http://www.schappellautomation.com/. [Accessed: 12-Mar-2018].

[60] J. Marrotte and A. Dixon, "Self-driving cars," 2017.

[61] H. R. Widditsch, "SPURV-The First Decade," *APL-UW 7215, Appl. Phys. Lab. Univ. Washingt.*, p. 32, 1973.

[62] T. E. Ewart, "Observations from straightline isobaric runs of SPURV," *Proc. IAPSO/IAMAP PSII*, pp. 1–18, 1976.

[63] "SPURV I." [Online]. Available: https://www.thinglink.com/scene/892853683962773504. [Accessed: 12-Mar-2018].

[64] J. G. Bellingham, C. A. Goudey, T. R. Consi, and C. Chryssostomidis, "A small, long-range autonomous vehicle for deep ocean exploration," in *The Second International Offshore and Polar Engineering Conference*, 1992.

[65] "Épaulard." [Online]. Available: https://en.wikipedia.org/wiki/Épaulard#/media/File:Ifremer_mg_5936.jpg. [Accessed: 12-Mar-2018].

[66] "Advanced Unmanned Search System (AUSS)." [Online]. Available: https://fas.org/irp/program/collect/auss.htm. [Accessed: 12-Mar-2018].

[67] "Ocean Explorer." [Online]. Available: https://oceanexplorer.noaa.gov/explorations/10chile/background/plan/media/missionplan3.html. [Accessed: 12-Mar-2018].

[68] "Odyssey." [Online]. Available: http://khoahoc.tv/odyssey-iv-truc-thang-duoi-long-bien-22942. [Accessed: 12-Mar-2018].

[69] T. Curtin, J. Bellingham, J. Catipovic, and D. Webb, "Autonomous Oceanographic Sampling Networks," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.

[70] B. Butler and M. R. Black, "The Theseus Autonomous Underwater Vehicle Two Successful Missions," in *INTERNATIONAL SYMPOSIUM ON*

UNMANNED UNTETHERED SUBMERSIBLE TECHNOLOGY, 1997, pp. 12–22.

[71] C. von Alt, B. Allen, T. Austin, and R. Stokey, "Remote environmental measuring units," in *Autonomous Underwater Vehicle Technology, 1994. AUV'94., Proceedings of the 1994 Symposium on*, 1994, pp. 13–19.

[72] G. Griffiths, K. G. Birch, N. W. Millard, S. D. McPhail, P. Stevenson, M. Pebody, J. R. Perrett, A. T. Webb, M. Squires, and A. Harris, "Oceanographic surveys with a 50 hour endurance autonomous underwater vehicle," in *Offshore Technology Conference*, 2000.

[73] C. Von Alt, "Autonomous underwater vehicles," *Auton. Underw. Lagrangian Platforms …*, pp. 1–5, 2003.

[74] H. Eisenbeiss, "A Mini Unmanned Aerial Vehicle (UAV): System overview and image acquisition," in *International Workshop on "Processing and Visualization using High-Resolution Imagery,"* 2004.

[75] Hui-Min Huang, "Autonomy Levels for Unmanned Systems (ALFUS) Framework Volume I : Terminology National Institute of Standards and Technology," *NIST Spec. Publ. 1011-I-2.0*, vol. I, no. October, pp. 0–46, 2008.

[76] D. J. DUDEK and J. B. WIENER, "General Distribution," *Econ. Policy*, no. 95, pp. 1–60, 1995.

[77] H. Bendea, P. Boccardo, S. Dequal, F. G. Tonolo, D. Marenchino, and M. Piras, "Low cost UAV for post-disaster assessment," *Proc. XXI Congr. Int. Soc. Photogramm. Remote Sens. Beijing China 311 July 2008*, vol. XXXVII, pp. 1373–1380, 2008.

[78] J. Cosic, P. Curkovic, J. Kasac, and J. Stepanic, "Interpreting Development of Unmanned Aerial Vehicles using Systems Thinking," *Interdiscip. Descr. Complex Syst.*, vol. 11, no. 1, pp. 143–152, 2013.

[79] S. G. Gupta, M. M. Ghonge, and P. M. Jawandhiya, "Review of Unmanned Aircraft System," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 2, no. 4, pp. 2278–1323, 2013.

[80] J. . Everaerts, "The use of unmanned aerial vehicles (uavs) for remote sensing and mapping," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XXXVII, no. Part B1, pp. 1187–1192, 2008.

[81] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegl, "Current Status and Future Perspectives for Unmanned Aircraft System Operations in the US," *J. Intell. Robot. Syst.*, vol. 52, no. 2, pp. 313–329, 2008.

[82] M. H. Fleming, S. J. Brannen, A. G. Mosher, B. Altmire, A. Metrick, M. Boyle, and R. Say, "Unmanned Systems in Homeland Security," *Homel. Secur. Stud. Anal. Inst.*, no. January, 2015.

[83] "As Dificuldade de Operações de Aeronaves Remotamente Pilotadas."

[Online]. Available: http://www.defesanet.com.br/vant/noticia/14998/As-Dificuldade-de-Operacoes-de-Aeronaves-Remotamente-Pilotadas-/. [Accessed: 12-Mar-2018].

[84] "MQ-8B Fire Scout." [Online]. Available: http://www.northropgrumman.com/Photos/pgL_MQ-10027_021.jpg. [Accessed: 12-Mar-2018].

[85] M. Dunbabin, A. Grinham, and J. Udy, "An autonomous surface vehicle for water quality monitoring," *Australas. Conf. Robot. Autom.*, pp. 2–4, 2009.

[86] R. Stelzer, "Robotic sailing: Overview," *OGAI J. (Oesterreichische Gesellschaft fuer Artif. Intell.*, vol. 27, no. 2, pp. 2–3, 2008.

[87] Scott Savitz, Irv Blickstein, Peter Buryk, Robert W. Button, Paul DeLuca, James Dryden, L. T. Jason Mastbaum, Jan Osburg, Philip Padilla, Amy Potter, Carter C. Price, and J. M. Y. Susan K. Woodward, Roland J. Yardley, *U.S. Navy employment options for unmanned surface vehicles (USVs)*. 2013.

[88] C. on A. V. in S. of Naval and Operations, *AUTONOMOUS VEHICLES IN SUPPORT OF NAVAL OPERATIONS*. THE NATIONAL ACADEMIES PRESS Washington, D.C., 2005.

[89] V. Bertram, "Unmanned Surface Vehicles – A Survey," in *Skibsteknisk Selskab, Copenhagen, Denmark*, 2008, pp. 1–14.

[90] M. M. Marques, R. Santos Carapau, A. V. Rodrigues, V. Lobo, J. Gouveia-Carvalho, W. Antunes, T. Gonçalves, F. Duarte, and B. Verissimo, "GammaEx project: A solution for CBRN remote sensing using Unmanned Aerial Vehicles in maritime environments," in *MTS/IEEE OCEANS 2017, Anchorage*, 2017, pp. 1–6.

[91] D. W. Gage, "UGV History 101: A Brief History of Unmanned Ground Vehicle Development Efforts," *Unmanned Syst. Mag.*, vol. 13, no. 3, 1995.

[92] A. K. Rajinder Kaur, "Unmanned Ground Vehicle (UGV)," *Oshkosk Def.*, vol. 4, no. 5, pp. 868–871, 2013.

[93] P.-N. Nguyen-Huu and J. Titus, "Reliability and Failure in Unmanned Ground Vehicle (UGV)," 2009.

[94] J. Ebken, M. Bruch, and J. Lum, "Applying unmanned ground vehicle technologies to unmanned surface vehicles," *Def. Secur.*, pp. 585–596, 2005.

[95] M. Ghaffari, S. M. Alhaj Ali, V. Murthy, X. Liao, J. Gaylor, and E. L. Hall, "Design of an unmanned ground vehicle, Bearcat III, theory and practice," *J. Robot. Syst.*, vol. 21, no. 9, pp. 471–480, 2004.

[96] "25 Cutting Edge Military Vehicles." [Online]. Available: https://yeahmotor.com/military/25-cutting-edge-military-vehicles/. [Accessed: 12-Mar-2018].

[97] D. Edison Thurman Hudson, Stephen Carl Licht and Patrick Eickstedt, “Unmanned Underwater Vehicle,” 2008.

[98] M. Caccia, G. Indiveri, and G. Veruggio, “Modeling and identification of open frame variable configuration unmanned underwater vehicles,” *IEEE J. Ocean. Eng.*, vol. 25, no. 2, pp. 227–240, 2000.

[99] A. V. Inzartsev, *Underwater Vehicles*. 2009.

[100] M. M. Maia, P. Soni, and F. J. Diez, “Demonstration of an Aerial and Submersible Vehicle Capable of Flight and Underwater Navigation with Seamless Air-Water Transition,” p. 9, 2015.

[101] B. Yamauchi and P. Rudakevych, “Griffon : A Man-Portable Hybrid UGV / UAV,” vol. 31, no. 5, pp. 443–450, 2004.

[102] P. L. J. Drews, A. A. Neto, and M. F. M. Camfile:///C:/Users/cinav/Desktop/Tese/6_References/CAP II/2.4 Unmanned Systems/Yamauchi, 2004.pdfpos, “Hybrid Unmanned Aerial Underwater Vehicle: Modeling and simulation,” *IEEE Int. Conf. Intell. Robot. Syst.*, no. April 2016, pp. 4637–4642, 2014.

[103] B. Fletcher, “Autonomous Vehicles and the Net-Centric Battlespace,” *Int. Unmanned Undersea Veh. Symp.*, p. 7, 2000.

[104] “The Navy Unmanned Undersea Vehicle (UUV) Master Plan,” 2004.

[105] R. T. Schnoor, “Modularized Unmanned Vehicle Packages for the Littoral Combat Ship Modularized Unmanned Vehicle Packages for the Littoral Combat Ship Mine Countermeasures Missions,” in *Oceans 2003 MTS/IEEE Conference*, 2003, vol. 298, no. 704, pp. 1–3.

[106] R. W. Button, J. Kamp, T. B. Curtin, and J. Dryden, *A Survey of Missions for Unmanned Undersea Vehicles*. Pittsburg: RAND Corporation, 2009.

[107] M. M. Marques, M. Gatta, M. Barreto, V. Lobo, A. Matos, B. Ferreira, J. Santos, P. Felisberto, S. Jesus, F. Zabel, R. Mendonça, and F. Marques, “Assessment of a shallow water area in the Tagus estuary using Unmanned Underwater Vehicle (or AUV â€™ s), vector- sensors , Unmanned Surface Vehicles , and Hexacopters –,” in *Proceedings of IEEE/MTS OCEANS 2018, Kobe*, 2018, pp. 1–5.

[108] B. T. Skrzypietz, “Unmanned Aircraft Systems for Civilian Missions,” *Brand. Inst. Soc. Secur. Policy Pap.*, no. 1, pp. 1–28, 2012.

[109] C. Nehme, J. W. Crandall, and M. L. Cummings, “An Operator Function Taxonomy for Unmanned Aerial Vehicle Missions,” *2th Int. Command Control Res. Technol. Symp.*, 2007.

[110] B. S. Sterling and C. W. Lickteig, “Command and control planning and teamwork: Exploring the future,” *Control*, 2000.

[111] J. Gouveia-Carvalho, W. Antunes, T. Gonçalves, M. M. Marques, and V. Lobo, "Unmanned Aerial Vehicles in chemical, biological, radiological and nuclear environments - Sensors review and concepts of operations," in *13 IARP Workshop on Humanitarian Demining and Similar Risky Interventions, HUDEM 2015, Croacia*, 2015, pp. 1–4.

[112] M. M. Marques, J. Gouvei-Carvalho, R. Pascoal, and C. Matos, "ATEX legal and standard framework applied to UAS in Mine Action and other risky interventions," in *14 IARP Workshop on Humanitarian Demining and Similar Risky Interventions, HUDEM 2016, Croacia*, 2016.

[113] R. Austin, *Unmanned aircraft systems*, no. April. 2007.

[114] M. M. Marques, V. Lobo, A. Salgado, M. Carreras, J. Roca, C. Candela, A. Martins, B. Ferreira, C. Almeida, E. Silva, F. A. De Sa, R. S. Carapau, P. Navy, and U. De Girona, "STRONGMAR Summer School 2016 – Joining theory with a practical application in Underwater Archeology," in *Proceedings of IEEE/MTS OCEANS 2017, Alberdeen*, 2016, pp. 1–6.

[115] D. Serrano, P. Chrobocinski, G. De Cubber, D. D. Moore, G. G. Leventakis, and S. Govindaraj, "ICARUS and DARIUS approaches towards interoperability," in *8th IARP Workshop on Robotics for Risky Environments*, 2015, pp. 1–12.

[116] H. Balta, G. De Cubber, Y. Baudoin, and D. Doroftei, "UAS deployment and data processing during the Balkans flooding with the support to Mine Action," in *8th IARP Workshop on Robotics for Risky Environments*, 2015, pp. 1–6.

[117] M. M. Marques, V. Lobo, R. Batista, J. Almeida, M. de F. Nunes, R. Ribeiro, and A. Bernardino, "Oil Spills Detection: Challenges addressed in the scope of the SEAGULL project," in *MTS/IEEE OCEANS Monterey 2016*, 2016, pp. 1–6.

[118] A. Bhardwaj, L. Sam, Akanksha, F. J. Martín-Torres, and R. Kumar, "UAVs as remote sensing platform in glaciology: Present applications and future prospects," *Remote Sens. Environ.*, vol. 175, pp. 196–204, 2016.

[119] B. Argrow, D. Lawrence, and E. Rasmussen, "Uav systems for sensor dispersal, telemetry, and visualization in hazardous environments," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005, p. 1237.

[120] "NATO. STANAG 4586, Edition No 3, Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability." [Online]. Available: http://nso.nato.int/nso/zPublic/stanags/current/4586eed03.pdf. [Accessed: 20-Mar-2015].

[121] NSA, "STANAG 4586 (EDITION 3): Standard Interfaces of UAV control systems (UCS) for NATO UAV interoperability," 2012.

[122] S. Frazzetta and M. Pacino, "A STANAG 4586 oriented approach to UAS

navigation," *J. Intell. Robot. Syst.*, vol. 69, no. 1–4, pp. 21–31, 2013.

[123] M. M. Marques, G. C. Rosa, F. Coito, and V. Lobo, "Two Major Architectures for Unmanned Systems – STANAG 4586 and JAUS," in *International Conference on Informatics, Control and Automation, Phuket*, 2015, vol. 4586, no. Stanag 4586, pp. 1–6.

[124] G. Feitshans, A. Rowe, J. Davis, M. Holland, and L. Berger, "Vigilant Spirit Control Station (VSCS): The Face of COUNTER," *AIAA Guid. Navig. Control Conf. Exhib.*, 2008.

[125] J. Pedersen, "A Practical View and Future Look at JAUS," 2006.

[126] H. I. Christensen and A. Hedstr, "STANAG - JAUS Study," 2004.

[127] R. Cuadrado, P. Royo, C. Barrado, M. Pérez, and E. Pastor, "Architecture issues and challenges for the integration of rpas in non-segregated airspace," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, pp. 1–11, 2013.

[128] R. S. Stansbury, M. A. Vyas, and T. A. Wilson, "A survey of UAS technologies for command, control, and communication (C3)," *J. Intell. Robot. Syst. Theory Appl.*, vol. 54, no. 1–3 SPEC. ISS., pp. 61–78, 2009.

[129] "JAUS Standard." [Online]. Available: https://www.sae.org/standards/. [Accessed: 10-Apr-2018].

[130] "OpenJAUS." [Online]. Available: http://openjaus.com. [Accessed: 21-Sep-2017].

[131] J. S. Wit, B. Drive, T. Air, and F. Base, "JOINT ARCHITECTURE FOR UNMANNED SYSTEMS (JAUS) TO SOCIETY OF AUTOMOTIVE ENGINEERS (SAE) TRANSITION October 2010," 2011.

[132] M. N. Clark, "JAUS compliant systems offers interoperability across multiple and diverse robot platforms," in *Proceedings of the AUVSI's Symposium Unmanned Systems North America (AUVSI'05)*, 2005, pp. 249–255.

[133] D. Erickson, "Standards for Representation in Autonomous Intelligent Systems," 2005.

[134] "JAUS Transport Considerations." [Online]. Available: https://www.sae.org/standards/content/air5645a/. [Accessed: 14-Apr-2018].

[135] "JAUS / SDP Transport Specification." [Online]. Available: https://www.sae.org/standards/content/as5669a/. [Accessed: 10-Apr-2018].

[136] "JAUS Messaging over the OMG Data Distribution Service (DDS)." [Online]. Available: https://www.sae.org/standards/content/arp6227/. [Accessed: 10-Apr-2018].

[137] "JAUS HMI Service Set." [Online]. Available: https://www.sae.org/standards/content/as6040/. [Accessed: 10-Apr-2018].

[138] "JAUS Compliance and Interoperability Policy." [Online]. Available: https://www.sae.org/standards/content/arp6012a/. [Accessed: 10-Apr-2018].

[139] "JAUS History and Domain Model." [Online]. Available: https://www.sae.org/standards/content/air5664a/. [Accessed: 10-Apr-2018].

[140] "JAUS Core Service Set." [Online]. Available: https://www.sae.org/standards/content/as5710a/. [Accessed: 10-Apr-2018].

[141] "JAUS Mobility Service Set." [Online]. Available: https://www.sae.org/standards/content/as6009a/. [Accessed: 10-Apr-2018].

[142] "JAUS Manipulator Service Set." [Online]. Available: https://www.sae.org/standards/content/as6057a/. [Accessed: 10-Apr-2018].

[143] "JAUS Service Interface Definition Language." [Online]. Available: https://www.sae.org/standards/content/as5684b/. [Accessed: 10-Apr-2018].

[144] "JAUS Unmanned Ground Vehicle Service Set." [Online]. Available: https://www.sae.org/standards/content/as6091/. [Accessed: 10-Apr-2018].

[145] "JAUS Mission Spooling Service Set." [Online]. Available: https://www.sae.org/standards/content/as6062/. [Accessed: 10-Apr-2018].

[146] "JAUS Environment Sensing Service Set." [Online]. Available: https://www.sae.org/standards/content/as6060/. [Accessed: 10-Apr-2018].

[147] R. Touchton, D. Kent, T. Galluzzo, C. D. Crane III, D. G. Armstrong II, N. Flann, J. Wit, and P. Adsit, "Planning and modeling extensions to the Joint Architecture for Unmanned Systems (JAUS) for application to unmanned ground vehicles," *Unmanned Gr. Veh. Technol. VII. Int. Soc. Opt. Photonics*, p. 146, 2005.

[148] A. Bahr, "Cooperative Localization for Autonomous Underwater Vehicles," 2009.

[149] W. T. Tsai, Y. Chen, and R. Paul, "Specification-based verification and validation of web services and service-oriented operating systems," in

Proceedings - International Workshop on Object-Oriented Real-Time Dependable Systems, WORDS, 2005, no. September, pp. 139–147.

[150] M. J. Hamilton, S. Kemna, and D. T. Hughes, "Antisubmarine warfare applications for autonomous underwater vehicles: The GLINT09 field trial results," *J. F. Robot.*, vol. 27, no. 6, pp. 890–902, 2010.

[151] J. Curcio, J. Leonard, and A. Patrikalakis, "SCOUT - A low cost autonomous surface platform for research in cooperative autonomy," *Proc. MTS/IEEE Ocean. 2005*, vol. 2005, 2005.

[152] B. Systems, "European Component Oriented Architecture (ECOA) Collaboration Programme," 2008. .

[153] P. M. Newman, "MOOS - Mission Orientated Operating Suite," 2005.

[154] S. Kemna, M. J. Hamilton, D. T. Hughes, and K. D. LePage, "Adaptive autonomous underwater vehicles for littoral surveillance," *Intell. Serv. Robot.*, vol. 4, no. 4, pp. 245–258, 2011.

[155] M. R. Benjamin and J. A. Curcio, "COLREGS-based navigation of autonomous marine vehicles," *2004 IEEE/OES Auton. Underw. Veh. (IEEE Cat. No.04CH37578)*, pp. 32–39, 2004.

[156] and M. D. G. R. P. Stokey, L. E. Freitag, "A Compact Control Language for AUV acustic communication," in *Oceans 2005-Europe*, 2005.

[157] H. Schneider, Toby and Schmidt, "The Dynamic Compact Control Language : A Compact Marshalling Scheme for Acoustic Communications," in *Oceans 2010 IEEE-Sydney*, 2010, pp. 1–10.

[158] T. Bean, G. Beidler, J. Canning, D. Odell, R. Wall, M. O. Rourke, M. Anderson, and D. Edwards, "Language and Logic to Enable Collaborative Behavior among Multiple Autonomous Underwater Vehicles," *Int. J. Intell. Syst.*, vol. 13, no. 1, pp. 67–80, 2008.

[159] R. P. Stokey, "A compact control language for autonomous underwater vehicles," *Woods Hole Oceanogr. Inst. Protoc.*, pp. 1133–1137, 2005.

[160] T. Schneider and H. Schmidt, "Unified Command and Control for Heterogeneous Marine Sensing Networks," *J. F. Robot.*, vol. 27, pp. 876--889, 2010.

[161] R. Duarte, Christiane N and Martel, Gerald R and Buzzell, Christine and Crimmins, Denise and Komerska, Rick and Mupparapu, Sai and Chappell, Steve and Blidberg, D Richard and Nitzel, "A Common Control Language to support multiple cooperating AUVs," in *Proceedings of the 14th International Symposium on Unmanned Untethered Submersible Technology*, 2005, pp. 1–9.

[162] C. N. Duarte, G. R. Martel, E. Eberbach, and C. Buzzell, "Talk amongst yourselves: getting multiple autonomous vehicles to cooperate," in *2004*

IEEE/OES Autonomous Underwater Vehicles, 2004, no. JULY.

[163] R. Martins, E. R. B. Marques, J. B. Sousa, P. S. Dias, J. Pinto, and F. L. Pereira, "IMC: A communication protocol for networked vehicles and sensors," in *Procedings IEEE /MTS OCEANS '09 Bremen*, 2009.

[164] E. Eberbach, C. Duarte, C. Buzzell, and G. Martel, "A portable language for control of multiple autonomous vehicles and distributed problem solving," in *Proc. of the 2nd Intern. Conf. on Computational Intelligence, Robotics and Autonomous Systems CIRAS*, 2003, pp. 15–18.

[165] S. S. Mupparapu, S. G. Chappell, R. J. Komerska, D. R. Blidberg, R. Nitzel, C. Benton, D. O. Popa, and A. C. Sanderson, "Autonomous systems monitoring and control (ASMAC) - an AUV fleet controller," in *Autonomous Underwater Vehicles, 2004 IEEE/OES*, 2004, pp. 119–126.

[166] R. and S. G. C. Komerska, "AUV Common Control Language (CCL) – A Proposed Standard Language and Framework for AUV Monitoring," 2007.

[167] E. Consortium, "EDA-NECSAVE," 2014.

[168] R. Diankov and J. Kuffner, "OpenRAVE : A Planning Architecture for Autonomous Robotics," *Robotics*, no. July, p. 34, 2008.

[169] R. Volpe, "Rover functional autonomy development for the mars mobile science laboratory," in *Proceedings IEEE Aerospace Conference*, 2003, vol. 2, pp. 643–652.

[170] R. Volpe, I. Nesnas, T. Estlin, and D. Mutz, "CLARAty: Coupled layer architecture for robotic autonomy," 2000.

[171] H. Volpe, Richard and Nesnas, Issa and Estlin, Tara and Mutz, Darren and Petras, Richard and Das, "The CLARAty architecture for robotic autonomy," in *Aerospace Conference, 2001, IEEE Proceedings*, 2001.

[172] I. A. D. Nesnas, R. Simmons, D. Gaines, C. Kunz, A. Diazcalderon, T. Estlin, R. Madison, J. Guineau, M. McHenry, I. H. Shu, and D. Apfelbaum, "CLARAty: Challenges and steps toward reusable robotic software," *Int. J. Adv. Robot. Syst.*, vol. 3, no. 1, pp. 023–030, 2006.

[173] I. Nesnas, "Claraty: A collaborative software for advancing robotic technologies," in *Proc. of NASA Science and Technology Conference*, 2007, pp. 1–7.

[174] I. A. D. Nesnas, "The CLARAty project: Coping with hardware and software heterogeneity," *Springer Tracts Adv. Robot.*, vol. 30, pp. 31–70, 2007.

[175] "ECOA." [Online]. Available: http://www.ecoa.technology/. [Accessed: 17-Apr-2017].

[176] M. Ababneh and J. M. Pullen, "Battle Management Language - Command

and control graphical user interface (BMLC2GUI)," in *Simulation Interoperability Workshop 2010,* 2010, pp. 337–348.

[177] A. Tolk and C. L. Blais, "Taxonomies, Ontologies, and Battle Management Languages – Recommendations for the Coalition BML Study Group," in *Simulation Interoperability Workshop*, 2005, no. April.

[178] K. Heffner and F. Hassaine, "Using BML for Command & Control of Autonomous Unmanned Air Systems," in *Simulation Interoperability Workshop, Orlando FL*, 2010.

[179] M. R. Hieb, U. Schade, and D. Fairfax, "Applying A Formal Language of Command and Control For Interoperability Between Systems," in *George Mason Univ Fairfax VA Center for Excellence in Command Control Communications Computers-Intelligence*, 2008.

[180] W. P. Sudnikovich, J. M. Pullen, M. S. Kleiner, and S. A. Carey, "Extensible Battle Management Language as a Transformation Enabler," *Simulation*, vol. 80, no. 12, pp. 669–680, 2004.

[181] M. R. Hieb and U. Schade, "Formalizing Command Intent Through Development of a Command and Control Grammar," in *12th International Command and Control Research and Technology Symposium*, 2007, pp. 1–20.

[182] D. T. Davis, "DESIGN, IMPLEMENTATION AND TESTING OF A COMMON DATA MODEL SUPPORTING AUTONOMOUS VEHICLE COMPATIBILITY AND INTEROPERABILITY," 2006.

[183] D. T. Davis, "AUTOMATED PARSING AND CONVERSION OF VEHICLE-SPECIFIC DATA INTO AUTONOMOUS VEHICLE CONTROL LANGUAGE (AVCL) USING CONTEXT-FREE GRAMMARS AND XML DATA BINDING Duane T . Davis , Naval Postgraduate School," in *Proceedings of the 14th International Symposium on Unmanned Untethered Submersible Technology*, 2005, no. August, pp. 1–11.

[184] R. Rasmussen and B. J. Hansen, "Experiment Report – SOA Pilot 2011 Keywords," 2012.

[185] S. S. Soh, "DETERMINING INTELLIGENCE, SURVEILLANCE AND RECONNAISSANCE (ISR) SYSTEM EFFECTIVENESS, AND INTEGRATION AS PART OF FORCE PROTECTION AND SYSTEM SURVIVABILITY," 2006.

[186] N.Group, "NIAG Study on Development of Conceptual Data Model for a Multi-Domain Unmanned Platform Control System (SG.202)," 2016.

[187] "ROS." [Online]. Available: http://www.ros.org/. [Accessed: 12-Apr-2016].

[188] G. C. C. e Silva, "Planeamento e Execução de Manobras de Veículos Aquáticos com Restrições de Movimento," FACULDADE DE

ENGENHARIA DA UNIVERSIDADE DO PORTO, 2013.

[189] K. A. Wyrobek, E. H. Berger, H. F. M. Van Der Loos, and J. K. Salisbury, “Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2008, pp. 2165–2170.

[190] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP Multiple Micro UAV Testbed,” *IEEE Robot. Autom. Mag.*, vol. 17, no. 3, pp. 56–65, 2010.

[191] S. T. John Leonard, Jonathan How, G. F. Mitch Berger, Stefan Campbell, A. H. Luke Fletcher, Emilio Frazzoli, Y. Sertac Karaman, Olivier Koch, E. O. Kuwata, David Moore, R. T. Steve Peters, Justin Teo, and and M. Walter, “A Perception-Driven Autonomous Urban Vehicle,” *J. F. Robot.*, p. 48, 2008.

[192] A. S. Huang, E. Olson, and D. C. Moore, “LCM: Lightweight Communications and Marshalling,” in *ProcedingsIEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010*, 2010, no. Lcm, pp. 4057–4062.

[193] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “the Pixhawk Open-Source Computer Vision Framework for Mavs,” in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2011, vol. XXXVIII-1/, no. September, pp. 13–18.

[194] A. S. Huang, M. Antone, E. Olson, L. Fletcher, D. Moore, S. Teller, and J. Leonard, “A High-Rate , Heterogeneous Data Set from the Darpa Urban Challenge,” *Int. J. Rob. Res.*, 2018.

[195] P. Schreiber, “Presentation robot advee,” *Eng. Mech.*, vol. 18, no. 5, pp. 307–322, 2011.

[196] H.-M. Huang, E. Messina, and J. Albus, “Autonomy Level Specification for Intelligent Autonomous Vehicles: Interim Progress Report,” in *2003 Performance Metrics for Intelligent Systems (PerMIS)*, 2003, no. September, pp. 1–7.

[197] C. Crick, G. Jay, S. Osentoski, B. Pitzer, and O. C. Jenkins, “Rosbridge: ROS for non-ROS users,” *Adv. Robot.*, vol. 100, pp. 493–504, 2017.

[198] “MAVlink.” [Online]. Available: http://qgroundcontrol.org/mavlink/start. [Accessed: 23-Oct-2016].

[199] H. Lim, J. Park, D. Lee, and H. J. Kim, “Build your own quadrotor,” *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 33–45, 2012.

[200] N. Reker, D. Troxell, and D. Troy, “Universal UAV Payload Interface,” California Polytechnic State University, 2015.

[201] L. Meier, P. Tanskanen, L. Heng, G. Hee, L. Friedrich, and F. M. Pollefeys, “PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight using

Onboard Computer Vision," *Int. Conf. Robot. Autom.*, vol. 231855, 2011.

[202] M. Coombes, O. McAree, W.-H. Chen, and P. Render, "Development of an autopilot system for rapid prototyping of high level control algorithms," in *Proceedings of 2012 UKACC International Conference on Control*, 2012, pp. 292–297.

[203] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Autonomous obstacle avoidance and maneuvering on a vision-guided MAV using on-board processing," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 2472–2477.

[204] M. Coombes, O. McAree, W.-H. Chen, and P. Render, "Development of a Generic Network Enabled Autonomous Vehicle System," in *Proceedings of 2014 UKACC International Conference on Control*, 2014, no. JANUARY, pp. 292–297.

[205] K. Shilov, "The Next Generation Design of Autonomous MAV Flight Control System SmartAP," 2014, pp. 225–229.

[206] J. a Marty, "Vulnerability Analysis of the MAVLink Protocol for Command and Control of Unmanned Aircraft," AIR FORCE INSTITUTE OF TECHNOLOGY, 2014.

[207] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. Marques, and J. Sousa, "The LSTS toolchain for networked vehicle systems," in *Procedings MTS/IEEE OCEANS 2013 Bergen*, 2013.

[208] A. Sanfeliu, N. Hagita, and A. Saffiotti, "Network robot systems," *Rob. Auton. Syst.*, vol. 56, no. 10, pp. 793–797, 2008.

[209] M. Duarte, J. Gomes, V. Costa, T. Rodrigues, F. Silva, V. Lobo, M. M. Marques, S. M. Oliveira, and A. L. Christensen, "Application of Swarm Robotic Systems to Marine Environmental Monitoring," in *Proceedings of IEEE/MTS OCEANS 2016, Xangai*, 2016, pp. 1–8.

[210] S. Khatib, *Handbook of Robotics*. Springer, 2008.

[211] C. Griffis and J. Schneider, "Unmanned Aircraft System Propulsion Systems Technology Survey," 2009.

[212] D. Cirigliano, "Engine - Type and Propulsion-Configuration Selections for Long-Duration UAV Flights," University of California, Irvine, 2017.

[213] N. J. Hobbs, "A Scalable Hybrid Power and Energy Architecture for Unmanned Ground Vehicles," The Pennsylvania State University The Graduate School College of Engineering, 2010.

[214] M. Dunbabin, B. Lang, and B. Wood, "Vision-based docking using an autonomous surface vehicle," in *Proceedings IEEE International Conference on Robotics and Automation 2008*, 2008, pp. 26–32.

[215] G. Elkaim and R. Kelbley, “Station Keeping and Segmented Trajectory Control of a Wind-Propelled Autonomous Catamaran,” *Proc. IEEE 45th Conf. Decis. Control*, pp. 2424–2429, 2006.

[216] J. A. Bowker, N. C. Townsend, M. Tan, and ..., “Experimental study of a wave energy scavenging system onboard autonomous surface vessels (ASVs),” in *Procedings IEEE /MTS OCEANS '15*, 2015, pp. 1–9.

[217] I. Ieropoulos, J. Greenman, and C. Melhuish, “Imitating Metabolism : Energy Autonomy in Biologically Inspired Robots,” in *Proceedings of the AISB '03, Second International Symposium on Imitation in Animals and Artifacts*, 2003, no. January 2003, pp. 1–4.

[218] F. du P. and W. C. Johan Meyer, *Design Considerations for Long Endurance Unmanned Aerial Vehicles*, vol. 2. 2018.

[219] R. O. Stroman, J. C. Kellogg, and K. Swider-lyons, “Testing of a PEM Fuel Cell System for Small UAV Propulsion Testing of a PEM Fuel Cell System for Small UAV Propulsion,” *Power*, no. July 2015, pp. 1–5, 2006.

[220] C.-B. M. Kweon, “A Review of Heavy-Fueled Rotary Engine Combustion Technologies,” 2011.

[221] G. Sutton and O. Biblarz, *Rocket Propulsion Elements*. 2012.

[222] W. Wenping Cao, B. C. Mecrow, G. J. Atkinson, J. W. Bennett, and D. J. Atkinson, “Overview of Electric Motor Technologies Used for More Electric Aircraft (MEA),” *IEEE Trans. Ind. Electron.*, vol. 59, no. 9, pp. 3523–3531, 2012.

[223] C. L. Griffis, T. A. Wilson, J. A. Schneider, and P. S. Pierpont, “Framework for the conceptual decomposition of unmanned aircraft propulsion systems,” in *Proceedings IEEE Aerospace Conference*, 2008.

[224] P. Szymak, “Mathematical model of underwater vehicle with undulating propulsion,” in *Proceedings - 2016 3rd International Conference on Mathematics and Computers in Sciences and in Industry, MCSI 2016*, 2017, pp. 269–274.

[225] E. De Margerie, J. B. Mouret, S. Doncieux, and J. A. Meyer, “Artificial evolution of the morphology and kinematics in a flapping-wing mini-UAV,” *Bioinspiration and Biomimetics*, vol. 2, no. 4, pp. 65–82, 2007.

[226] P. Deusdado, E. Pinto, M. Guedes, F. Marques, P. Rodrigues, A. Lourenço, R. Mendonça, A. Silva, P. Santana, J. Corisco, M. Almeida, L. Portugal, R. Caldeira, J. Barata, and L. Flores, “An aerial-ground robotic team for systematic soil and biota sampling in estuarine mudflats,” *Adv. Intell. Syst. Comput.*, vol. 418, pp. 15–26, 2016.

[227] R. Dasgupta and S. Dey, “A Comprehensive Sensor Taxonomy and Semantic Knowledge Representation,” in *Seventh International Conference*

on Sensing Technology, 2013, pp. 791–799.

[228] L. Xue, Y. Liu, P. Zeng, H. Yu, and Z. Shi, "An ontology based scheme for sensor description in context awareness system," in *Proceding IEEE International Conference on Information and Automation, ICIA 2015*, 2015, pp. 817–820.

[229] N. Chen, C. Hu, Y. Chen, C. Wang, and J. Gong, "Using SensorML to construct a geoprocessing e-Science workflow model under a sensor web environment," *Comput. Geosci.*, vol. 47, pp. 119–129, 2012.

[230] M. Botts, G. Percivall, C. Reed, and J. Davidson, "OGC Sensor Web Enablement: Overview and High Level Architecture," *Lecture Notes In Computer Science*, vol. 4540, no. December. pp. 175–190, 2007.

[231] R. Mautz and S. Tilch, "Optical Indoor Positioning Systems," *Proceding IEEE Int. Conf. 2011*, no. September, pp. 21–23, 2011.

[232] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," in *Procedings IEEE Computer August 2001*, 2001, no. August.

[233] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile Robot Positioning & Sensors and Techniques," *J. Robot. Syst. Spec. Issue Mob. Robot.*, vol. 14, no. 4, pp. 231–249, 1997.

[234] R. G. Folsom, "Review of the Pitot Tube," 1955.

[235] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[236] B. Alving, K. Gade, K. Svartveit, A. B. Willumsen, and R. Sørhage, "DVL Velocity Aiding in Hugin 1000 Integrated Inertial Navigation System.pdf," *Model. Identif. Control*, 2004.

[237] D. B. Barber, S. R. Griffiths, T. W. McLain, and R. W. Beard, "Autonomous Landing of Miniature Aerial Vehicles," *J. Aerosp. Comput. Information, Commun.*, vol. 4, no. 5, pp. 770–784, 2007.

[238] M. Caruso and T. Bratland, "A new perspective on magnetic field sensing," *Sensors*, vol. 15, pp. 34–47, 1998.

[239] P. Ripka and M. Janosek, "Advances in Magnetic Field Sensors," *IEEE Sens. J.*, vol. 10, no. 6, pp. 1108–1116, 2010.

[240] Y. K. Chan and V. C. Koo, "An Introduction to Synthetic Aperture Radar (SAR)," 2008.

[241] M. Skolnik, *An Introduction to RADAR*, vol. 1634. Radar Handbook, 1962.

[242] F. Amzajerdian, D. Pierrottet, L. Petway, G. Hines, and V. Roback, "Lidar systems for precision navigation and safe landing on planetary bodies," in *International Symposium on Photoelectronic Detection and Imaging 2011*, 2011,

vol. 2, p. 819202.

[243] P. Blondel and B. J. Murton, *Handbook of Seafloor Sonar Imagery*. 1997.

[244] M. Agrawal and K. Konolige, "Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS," in *18th International Conference Pattern Recognition*, 2006.

[245] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation," *Auton. Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[246] D. Barrick, *FM/CW Radar Signals and Digital Processing*, no. July 1973. 1973.

[247] L. L. Whitcomb, D. R. Yoerger, and H. Singh, "Combined Doppler/LBL based navigation of underwater vehicles," in *11th International Symposium on Unmanned Untethered Submersible Technology*, 1999, no. 10015, pp. 1–7.

[248] G. Lutz, *Semiconductor Radiation Detectors*. 2002.

[249] R. Park, "Thermocouple Fundamentals," 2010.

[250] Ladyada, "Thermistor," 2013.

[251] R. Martin, "Sensor Basics : Types , Functions and Applications," 2013.

[252] D. Sheingold, *Transducer Interfacing Handbook - A Guide to Analog Signal Conditioning*. 1978.

[253] A. B. Chatfield, *Fundamentals Of High Accuracy Inertial Navigation*. 1997.

[254] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications*, vol. 1, no. v. 1. 1996.

[255] J. Yuh, T. Ura, and G. Bekey, *Underwater Robots*, vol. 3, no. June. 1996.

[256] R. Bencatel, M. Faied, J. Sousa, and A. Girard, "Formation control with collision avoidance," in *IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 591–596.

[257] M. M. Marques, P. Dias, N. P. Santos, V. Lobo, R. Batista, D. Salgueiro, A. Aguiar, M. Costa, J. E. da Silva, A. S. Ferreira, others, J. Morgado, R. Batista, D. Salgueiro, R. Ribeiro, J. S. Marques, A. Bernardino, M. Griné, M. Taiana, A. S. Ferreira, and J. Sousa, "Unmanned Aircraft Systems in Maritime Operations: Challenges addressed in the scope of the SEAGULL project," in *OCEANS 2015-Genova*, 2015, pp. 1–6.

[258] Dixon S.R., Wickens C.D., and Chang D., "Mission Control of Multiple Unmanned Aerial Vehicles: A Workload Analysis," *Hum. Factors*, vol. 47, no. 3, pp. 479–487, 2005.

[259] N. Wang, N. Zhang, and M. Wang, "Wireless sensors in agriculture and food industry — Recent development and future perspective," *Comput. Electron. Agric.*, vol. 50, pp. 1–14, 2006.

[260] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey,"

Comput. Networks, vol. 52, no. 12, pp. 2292–2330, 2008.

[261] M. Wzorek, D. Land, P. Doherty, S.- Link, and D. Land, "GSM Technology as a Communication Media for an Autonomous Unmanned Aerial Vehicle," in *Proceedings of the 21st Bristol International UAV Systems Conference*, 2006, no. 21, p. 41.

[262] M. M. Marques, V. Lobo, R. Batista, J. Oliveira, A. P. Aguiar, J. E. Silva, J. B. de Sousa, M. de F. Nunes, R. A. Ribeiro, A. Bernardino, and J. S. Marques, "An unmanned aircraft system for maritime operations," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 4, p. 172988141878633, 2018.

[263] M. Draper, G. Calhoun, H. Ruff, D. Williamson, and T. Barry, "Manual versus speech input for unmanned aerial vehicle control station operations," in *47th Annual Meeting of the Human Factors and Ergonomics Society*, 2003, vol. 47, pp. 109–113.

[264] F. Segor, A. Bürkle, T. Partmann, and R. Schönbein, "Mobile ground control station for local surveillance," in *5th International Conference on Systems, ICONS 2010*, 2010, pp. 152–157.

[265] A. Pascoal, P. Oliveira, C. Silvestre, L. Sebasti??o, M. Rufino, V. Barroso, J. Gomes, G. Ayela, P. Coince, M. Cardew, A. Ryan, H. Braithwaite, N. Cardew, J. Trepte, N. Seube, J. Champeau, P. Dhaussy, V. Sauce, R. Moiti??, R. Santos, F. Cardigos, M. Brussieux, and P. Dando, "Robotic ocean vehicles for marine science applications: The european ASIMOV project," in *Oceans Conference Record (IEEE)*, 2000, vol. 1, no. FEBRUARY, pp. 409–415.

[266] M. Eriksson and P. Ringman, "Launch and recovery systems for unmanned vehicles onboard ships. A study and initial concepts.," 2013.

[267] "UAVs: Launch and recovery," *Air Sp. Eur.*, vol. 1, no. 5–6, pp. 59–62, 1999.

[268] P. Fahlstrom and T. Gleason, *Introduction to UAV Systems*. Wiley, 2012.

[269] R. Skulstad, C. Syversen, M. Merz, N. Sokolova, T. Fossen, and T. Johansen, "Autonomous net recovery of fixed-wing UAV with single-frequency carrier-phase differential GNSS," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 5, pp. 18–27, 2015.

[270] B. D. Reineman, L. Lenain, and W. K. Melville, "The use of ship-launched fixed-wing UAVs for measuring the marine atmospheric boundary layer and ocean surface processes," *J. Atmos. Ocean. Technol.*, vol. 33, no. 9, pp. 2029–2052, 2016.

[271] L. Hanyok and T. Smith, "Launch and Recovery System Literature Review," 2010.

[272] A. (BMT D. S. Kimber, "Boat Launch and Recovery - A Key Enabling Technology for Flexible Warships," in *Pacific 2012*, 2012, no. February.

[273] F. Morais, M. M. Marques, T. Ramalho, P. Sinogas, N. P. Santos, and V. Lobo, "Trajectory and Guidance Mode for autonomously landing an UAV on a naval platform using a vision approach," in *Proceedings of the IEEE/MTS OCEANS 2015, Genova*, 2015, pp. 1–7.

[274] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE Trans. Robot. Autom.*, vol. 19, no. 3, pp. 371–380, 2003.

[275] G.-J. Duan and P.-F. Zhang, "Research on Application of UAV for Maritime Supervision," *J. Shipp. Ocean Eng.*, vol. 4, pp. 322–326, 2014.

[276] Z. F. Yang and W. H. Tsai, "Using parallel line information for vision-based landmark location estimation and an application to automatic helicopter landing," *Robot. Comput. Integr. Manuf.*, vol. 14, no. 4, pp. 297–306, 1998.

[277] G. Xu, Y. Zhang, S. Ji, Y. Cheng, and Y. Tian, "Research on computer vision-based for UAV autonomous landing on a ship," *Pattern Recognit. Lett.*, vol. 30, no. 6, pp. 600–605, 2009.

[278] C. Fu, A. Carrio, M. A. Olivares-Mendez, R. Suarez-Fernandez, and P. Campoy, "Robust real-time vision-based aircraft Tracking from Unmanned Aerial Vehicles," *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 5441–5446, 2014.

[279] V. Khithov, A. Petrov, I. Tishchenko, and K. Yakovlev, "Toward autonomous UAV landing based on infrared beacons and particle filtering," *Adv. Intell. Syst. Comput.*, vol. 447, pp. 529–537, 2017.

[280] V. Lepetit and F. M. Pascal, "EPnP: An Accurate O(n) Solution to the PnP Problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, 2009.

[281] W. Kong, D. Zhang, X. Wang, Z. Xian, and J. Zhang, "Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 2963–2970.

[282] E. Altu, J. P. Ostrowski, and R. Mahony, "Control of a Quadrotor Helicopter Using Visual Feedback," in *International Conference on Robotics & Automation*, 2002, no. May, pp. 72–77.

[283] O. A. Yakimenko, I. I. Kaminer, W. J. Lentz, and P. A. Ghyzel, "Unmanned aircraft navigation for shipboard landing using infrared vision," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 4, pp. 1181–1200, 2002.

[284] A. Hazeldene, A. Sloan, C. Wilkin, and A. Price, "In-Flight Orientation , Object Identification and Landing Support for an Unmanned Air Vehicle," in *International Conference on Autonomous Robots and Agents*, 2004, pp. 333–338.

[285] M. M. Marques, P. Dias, J. Morgado, R. Batista, D. Salgueiro, R. Ribeiro, J.

S. Marques, A. Bernardino, M. Griné, M. Taiana, A. S. Ferreira, and J. Sousa, “Unmanned Aircraft Systems in Maritime Operations: Challenges addressed in the scope of the SEAGULL project,” in *Proceedings of the IEEE/MTS OCEANS 2015, Genova*, 2015, pp. 1–6.

[286] S. Freitas, C. Almeida, H. Silva, J. Almeida, and E. Silva, “Supervised classification for hyperspectral imaging in UAV maritime target detection,” *2018 IEEE Int. Conf. Auton. Robot Syst. Compet.*, pp. 84–90, 2018.

[287] G. P. Roussos, G. Chaloulos, K. J. Kyriakopoulos, and J. Lygeros, “Control of multiple non-holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation functions,” in *Proceedings of the IEEE Conference on Decision and Control*, 2008, no. May 2014, pp. 1225–1230.

[288] J. S. Marques, A. Bernardino, G. Cruz, and M. Bento, “An algorithm for the detection of vessels in aerial images,” *11th IEEE Int. Conf. Adv. Video Signal-Based Surveillance, AVSS 2014*, no. September, pp. 295–300, 2014.

[289] “PixHawk.” [Online]. Available: https://pixhawk.org/. [Accessed: 17-May-2017].

[290] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “PIXHAWK: A System for Autonomous Flight using Onboard Computer Vision,” in *International Conference on Robotics and Automation (ICRA) 2011*, 2011, pp. 2992–2997.

[291] C. M. Humphrey and J. A. Adams, “Robotic Tasks for CBRNE Incident Response,” *Adv. Robot.*, vol. 23, pp. 1217–1232, 2009.

[292] M. M. Marques, V. Lobo, J. Gouveia-carvalho, A. José, M. Nogueira, C. Matos, R. S. Carapau, and A. V. Rodrigues, “CBRN remote sensing using Unmanned Aerial Vehicles : Challenges addressed in the scope of the GammaEx project regarding hazardous materials and environments,” in *The 6th International Conference on Risk Analysis and Crisis Response (RACR-2017)*, 2017.

[293] D. Doroftei, A. Matos, and G. de Cubber, “Designing search and rescue robots towards realistic user requirements,” *Appl. Mech. Mater.*, vol. 658, pp. 612–617, 2014.

[294] M. L. Incze, S. R. Sideleau, C. Gagner, and C. A. Pippin, “Communication and collaboration among heterogeneous unmanned systems using SAE JAUS standard formats and protocols,” *IFAC-PapersOnLine*, vol. 28, no. 5, pp. 7–10, 2015.

[295] S. Govindaraj, K. Chintamani, J. Gancet, P. Letier, B. Van Lierde, Y. Nevatia, G. De Cubber, D. Serrano, M. Esbri Palomares, J. Bedkowski, C. Armbrust, J. Sanchez, A. Coelho, and I. Orbe, “The ICARUS project - Command, Control and Intelligence (C2I),” in *2013 IEEE International*

Symposium on Safety, Security, and Rescue Robotics, SSRR 2013, 2013.

[296] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, “Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2015, vol. 2015–June, no. June, pp. 6423–6430.

[297] G. Kruijff, I. Kruijff-korbayov, S. Keshavdas, M. Jan, F. Colas, M. Liu, R. Siegwart, M. Neerincx, and B. Larochelle, “Designing , developing , and deploying systems to support human – robot teams in disaster response,” *Adv. Robot.*, vol. 28, no. 23, pp. 1547–1570, 2014.

[298] B. M. Ferreira, A. C. Matos, and J. C. Alves, “Water-jet propelled autonomous surface vehicle UCAP: System description and control,” in *Proceedings of the IEEE/MTS OCEANS 2016, Shanghai*, 2016, pp. 4–8.

[299] A. Matos, E. Silva, N. Cruz, J. C. Alves, D. Almeida, M. Pinto, A. Martins, J. Almeida, and D. Machado, “Development of an Unmanned Capsule for large-scale maritime search and rescue,” *Proc. IEEE/MTS Ocean. 2013*, pp. 1–8, 2013.

[300] D. Doroftei, A. Matos, E. Silva, V. Lobo, R. Wagemans, and G. De Cubber, “Operational validation of robots for risky environments,” in *8th IARP Workshop on Robotics for Risky Environments*, 2015.

[301] M. M. Marques, A. Martins, A. Matos, N. Cruz, J. M. Almeida, J. C. Alves, V. Lobo, and E. Silva, “REX 2014-Robotic Exercises 2014 Multi-robot field trials,” in *Proceedings of the IEEE/MTS 2015, Washington*, 2015.

[302] G. Ferri, F. Ferreira, V. Djapic, Y. Petillot, M. P. Franco, and A. Winfield, “The euRathlon 2015 grand challenge: The first outdoor multi-domain search and rescue robotics competition— A Marine perspective,” *Mar. Technol. Soc. J.*, vol. 50, no. 4, pp. 81–97, 2016.

[303] M. M. Marques, R. Parreira, V. Lobo, A. Martins, A. Matos, N. Cruz, J. M. Almeida, J. C. Alves, E. Silva, J. Będkowski, K. Majek, M. Pełka, P. Musialik, H. Ferreira, A. Dias, B. Ferreira, G. Amaral, A. Figueiredo, R. Almeida, F. Silva, D. Serrano, G. Moreno, G. De Cubber, H. Balta, and H. Beglerović, “Use of multi-domain robots in search and rescue operations - Contributions of the ICARUS team to the euRathlon 2015 challenge,” in *Proceedings of the IEEE/MTS OCEANS 2016, Shanghai*, 2016.

[304] A. V. Rodrigues, “Implementação de um tradutor entre STANAG 4586 e MAVLink,” Portruguese Naval Academy, 2017.

[305] A. V. Rodrigues, R. S. Carapau, M. M. Marques, V. Lobo, and F. Coito, “Unmanned systems interoperability in military maritime operations: MAVLink to STANAG 4586 bridge,” in *Proceedings of the IEEE/MTS OCEANS 2017, Aberdeen*, 2017, vol. 2017–Octob, pp. 1–5.

[306] S. Freitas, H. Silva, J. Almeida, and E. Silva, "Hyperspectral Imaging for Real-Time Unmanned Aerial Vehicle Maritime Target Detection," *J. Intell. Robot. Syst. Theory Appl.*, vol. 90, no. 3–4, pp. 551–570, 2018.

[307] H. Silva, J. M. Almeida, F. Lopes, J. P. Ribeiro, S. Freitas, G. Amaral, C. Almeida, A. Martins, and E. Silva, "UAV trials for multi-spectral imaging target detection and recognition in maritime environment," in *Proceedings IEEE/MTS OCEANS 2016*, 2016.

[308] J. P. Ribeiro, H. Fontes, M. Lopes, H. Silva, R. Campos, J. M. Almeida, and E. Silva, "UAV cooperative perception based on DDS communications network," in *Proceedings IEEE/MTS OCEANS 2017 - Anchorage*, 2017.

[309] V. Rooij, C. A. T. Patric, N. Daugherty, and P. C. A. T. Van Rooij, "What is the future for SOF in the Arctic ?," Naval Postgraduate School, 2014.

[310] "NORTH ATLANTIC TREATY ORGANIZATION SCIENCE AND TECHNOLOGY ORGANIZATION." [Online]. Available: https://www.sto.nato.int/Pages/organization.aspx.